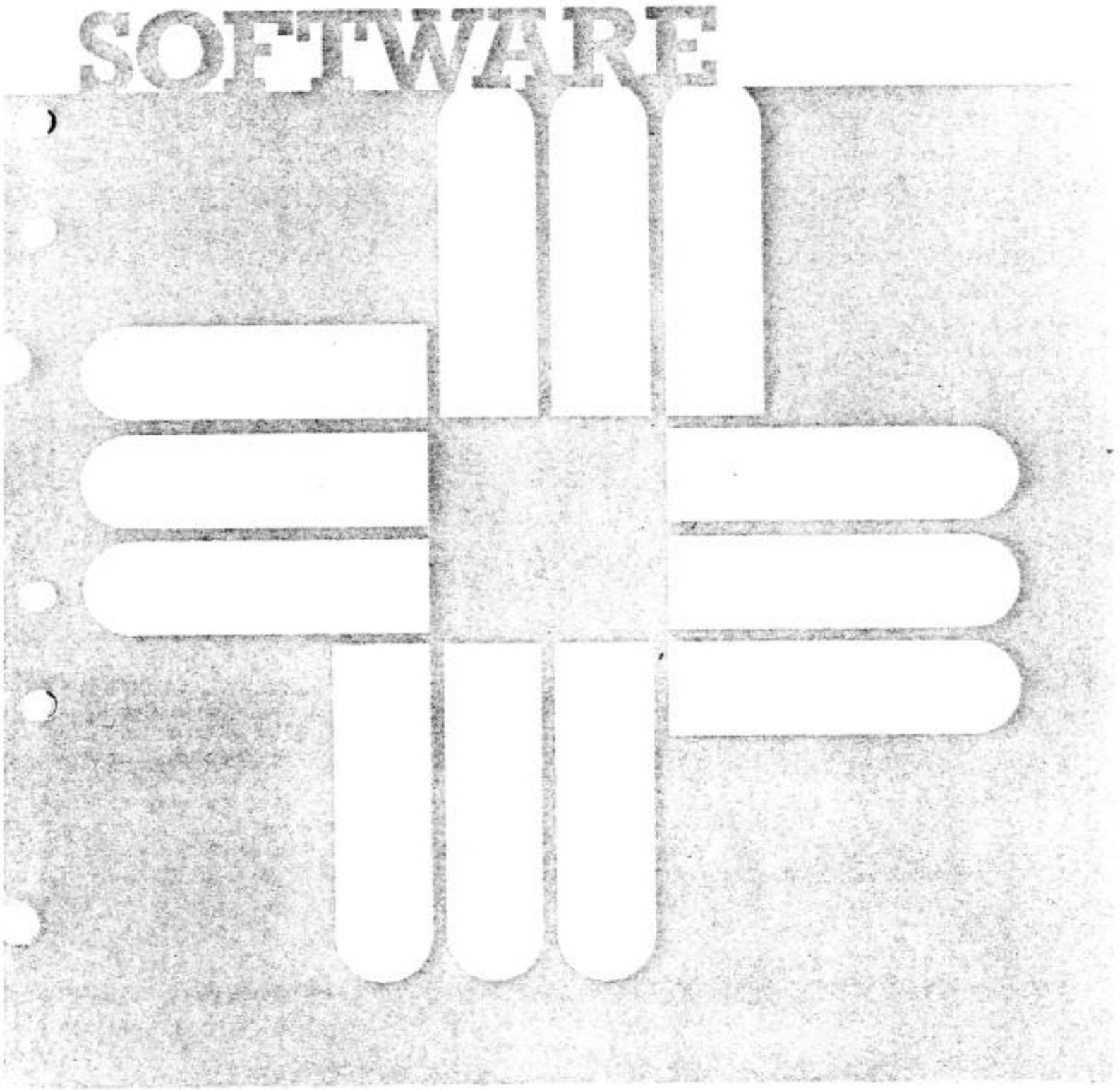
MODULAR COMPUTER SYSTEMS, INC.

MAX IV System Guide Manual

Unit Record Device Handlers



MANUAL HISTORY

Manual Order Number: 213-804006-I01

Title: MAX IV SYSTEM GUIDE MANUAL, UNIT RECORD DEVICE HANDLERS

Model Number: 8240I

Revision Level	Date Issued	Description
H00	03/83	Initial issue (H.0). (This manual replaces chapters 3, 4, 5, 6, 7, 101, and 102 of the MAX IV INPUT/OUTPUT Reference Manual (210-610501-000).
H02	07/84	Revision (H.2). Changes made to Chapter 10, the MAX IV Imaginary Printer Spooling Symbiont.
I00	09/84	Reissue (I.0). Lowercase to uppercase roll option added to line printer handler, LP.HAN
I01	08/85	Revision (I.1). UO option added to spooler.

Contents subject to change without notice.

Copyright © 1983, by Modular Computer Systems, Inc.
All Rights Reserved.
Printed in the United States of America.

PREFACE

Audience: The information in this manual is directed to the applications programmer or developer with a good knowledge of MODCOMP Assembler language and familiarity with MAX IV SYSGEN and REX Services.

Subject: This manual contains information about the MAX IV Unit Record Device Handlers. The various I/O operations and modes are described. . Error/Event indicators and System Generation information are provided.

Those users familiar with previous issues of MAX IV manuals may notice that the information contained in the MAX IV INPUT/OUTPUT, Reference Manual, 210-610501-000 has been reorganized to provide easier accessibility.

The information contained in the chapters listed below is now available as this MAX IV UNIT RECORD DEVICE HANDLERS, System Guide Manual.

Chapter 3	MAX IV TELETYPEWRITER, CRT, PAPER-TAPE READER/PUNCH HANDLER
Chapter 4	MAX IV CARD READER/PUNCH HANDLER
Chapter 5	MAX IV LINE PRINTER HANDLER
Chapter 6	MAX IV RASTER PRINTER/PLOTTER HANDLER
Chapter 7	MAX IV INCREMENTAL X-Y PLOTTER HANDLER
Chapter 101	MAX IV IMAGINARY PRINTER SPOOLING SYMBIONT
Chapter 102	MAX IV INPUT SPOOLING SYMBIONT

Products Supported: MAX IV Software System, Model 8240I.

Special Symbols and Notations: A revision bar (|) located in the margin of the page in the text indicates a change to the manual. This change generally represents a technical change to the product due to product revision. A revision bar is also entered in the Table of Contents to flag the general location of changes in the text.

Related Publications: The reader is referred to the manuals listed below for additional information.

Manual Order Number	Manual Title
213-804001-REV System Guide Manual	MAX IV GENERAL OPERATING SYSTEM
213-804002-REV System Guide Manual	MAX IV SYSGEN
213-804003-REV System Guide Manual	MAX IV EXECUTIVE (REX) SERVICES
213-804004-REV System Guide Manual	MAX IV OPERATOR COMMUNICATIONS
213-804005-REV System Guide Manual	MAX IV BASIC INPUT/OUTPUT SYSTEM
213-804007-REV System Guide Manual	MAX IV DATA STORAGE DEVICE HANDLERS
213-804008-REV System Guide Manual	MAX IV COMMUNICATIONS HANDLERS

When ordering manuals, use the Manual Order Number listed in this Preface. The latest revision (REV) will be shipped.

MAX IV REVISION I.1 SUMMARY

Spooler

An UO option was added to give the spooler exclusive use of a physical device until the end of a print file.

MAX IV REVISION I.Ø SUMMARY

Line Printer Handler - LP.HAN

The ability to print files containing lowercase characters on an uppercase only printer has been added to LP.HAN. This uppercase roll option is selected at SYSGEN on the DEVICE statement. When selected, lowercase characters are converted into their corresponding uppercase counterparts before they are sent to the printer.

Raster Printer/Plotter Handler - VP.HAN

Additional information was added to VP.HAN in the system generation (SYSGEN) section in response to an SPR.

TABLE OF CONTENTS

		Page
CHAPTER 1	INTRODUCTION	1
1.1	HANDLER'S RESPONSE	1
1.2	DEVICE DEPENDENT FEATURES	2
1.3	HANDLER SYSGEN	2
1.4	HOLDUP CONDITIONS	2
1.5	SPOOLING SYMBIONTS	2
CHAPTER 2	MAX IV HALF-DUPLEX TELETYPEWRITER/CRT HANDLER (TYn)	3
2.1	STANDARD CONTROLLERS AND DEVICES	3
2.2	I/O OPERATIONS AND MODES	4
2.2.1	STANDARD ASCII READ FROM KEYBOARD (TYn)	4
2.2.2	NON-STANDARD ASCII READ FROM KEYBOARD (TYn)	6
2.2.3	BINARY READ FROM KEYBOARD (TYn)	7
2.2.4	STANDARD ASCII WRITE TO PRINTER (TYn)	7
2.2.5	NON-STANDARD ASCII WRITE TO PRINTER (TYn)	8
2.2.6	FILE/RECORD OPERATIONS	9
2.3	SPECIAL QUEUING RULES	10
2.4	UFT OPTIONS	11
2.4.1	DEVICE-DEPENDENT OPTION	12
2.4.2	INHIBIT DATA EDITING	12
2.4.3	TERMINATOR BYTE	12
2.5	HOLDUP CONDITIONS REPORTED TO OPERATOR	13
2.6	SYSTEM GENERATION	13
2.6.1	STANDARD PARAMETERS	13
2.6.2	DEVICE OPTION PARAMETER	14
CHAPTER 3	MAX IV PAPER TAPE READER HANDLER (PRn)	17
3.1	I/O OPERATIONS AND MODES	17
3.1.1	ASCII READ FROM PAPER TAPE READER (PRn)	17
3.1.2	BINARY READ FROM PAPER TAPE READER (PRn)	18
3.1.3	ADVANCE RECORD FROM PAPER TAPE READER (PRn)	21
3.1.4	ADVANCE FILE FROM PAPER TAPE READER (PRn)	21
3.1.5	FILE/RECORD OPERATIONS	21
3.2	UFT OPTIONS	22
3.2.1	DEVICE-DEPENDENT OPTIONS	22
3.3	HOLDUP CONDITIONS REPORTED TO OPERATOR	23
3.4	SYSTEM GENERATION	23

CHAPTER 4 MAX IV PAPER TAPE PUNCH HANDLER (TPn,PPn) 27

4.1 I/O OPERATIONS AND MODES 27

4.1.1 ASCII WRITE TO PUNCH (TPn,PPn) 27
4.1.2 BINARY WRITE TO PUNCH (TPn,PPn) 29
4.1.3 FILE/RECORD OPERATIONS 30

4.2 HOLDUP CONDITIONS REPORTED TO OPERATOR 31
4.3 SYSTEM GENERATION 31

CHAPTER 5 MAX IV CARD READER HANDLER (CRn/DRn) 35

5.1 A BRIEF DESCRIPTION OF HANDLER OPERATION 35
5.2 I/O OPERATIONS AND MODES 38

5.2.1 STANDARD ASCII READ FROM CARD READER (CRn) .. 38
5.2.2 NON-STANDARD ASCII READ FROM CARD READER (CRn) 40
5.2.3 STANDARD BINARY READ FROM CARD READER (CRn) . 40
5.2.4 NON-STANDARD BINARY READ FROM CARD READER (CRn) 41
5.2.5 FILE/RECORD OPERATIONS 41

5.3 ERROR RECOVERY AND HOLDUP CONDITIONS 42
5.4 SYSTEM GENERATION 43

CHAPTER 6 MAX IV CARD PUNCH HANDLER (CPn) 47

6.1 I/O OPERATIONS AND MODES 47

6.1.1 ASCII WRITE TO CARD PUNCH (CPn) 47
6.1.2 BINARY WRITE TO CARD PUNCH (CPn) 48
6.1.3 OTHER OPERATIONS 48

6.2 SPECIAL UNIMPLEMENTED HARDWARE FEATURE 50
6.3 ERROR RECOVERY AND HOLDUP CONDITIONS 50
6.4 SYSTEM GENERATION 51

CHAPTER 7 MAX IV LINE PRINTER HANDLER (LPn, LCn, DPn) 55

7.1 I/O OPERATIONS AND MODES 56

7.1.1 STANDARD ASCII WRITE TO PRINTER (LPn, LCn, DPn) 56
7.1.2 NON-STANDARD ASCII WRITE TO PRINTER (LPn, LCn, DPn) 57
7.1.3 BINARY WRITE TO PRINTER 57
7.1.4 FILE/RECORD OPERATIONS 57
7.1.5 DEFERRED PRINTING 58

7.2 ERROR RECOVERY AND HOLDUP CONDITIONS 58
7.3 PRINTER QUALIFICATION 59

7.4 SYSTEM GENERATION 59

7.4.1 STANDARD PARAMETERS 59
7.4.2 DEVICE OPTION PARAMETER 60

CHAPTER 8 MAX IV RASTER PRINTER/PLOTTER HANDLER (VPn) 63

8.1 I/O OPERATIONS AND MODES 63

8.1.1 STANDARD ASCII WRITE TO PRINTER (VPn) 63
8.1.2 NON-STANDARD ASCII WRITE TO PRINTER (VPn) ... 64
8.1.3 BINARY WRITE TO PLOTTER (VPn) 65
8.1.4 FILE/RECORD OPERATIONS 65
8.1.5 DEFERRED PRINTING 66

8.2 UFT OPTIONS 67

8.2.1 SIMULTANEOUS PRINT/PLOT CAPABILITY 67

8.3 ERROR RECOVERY AND HOLDUP CONDITIONS 67
8.4 SYSTEM GENERATION 68

CHAPTER 9 MAX IV INCREMENTAL X-Y PLOTTER HANDLER (XYn) 71

9.1 I/O OPERATIONS AND MODES 71

9.1.1 ASCII AND STANDARD BINARY 71
9.1.2 BINARY WRITE TO PLOTTER (XYn) 71
9.1.3 FILE/RECORD OPERATIONS 72

9.2 HOLDUP CONDITIONS REPORTED TO OPERATOR 73
9.3 SYSTEM GENERATION 74

CHAPTER 10 MAX IV IMAGINARY PRINTER SPOOLING SYMBIONT 77

10.1 SPOOLING AND THE PROGRAMMER 77
10.2 SPOOLING SYMBIONT 77
10.3 IMAGINARY DEVICE SECTION OPERATION 78
10.4 PHYSICAL DEVICE SECTION OPERATION 81
10.5 OPTIONAL DESIGN FEATURES 82
10.6 INSTALLING THE PRINTER SPOOLER 83

10.6.1 TASK PREPARATION 83
10.6.2 CATALOGING A NON-RESIDENT SPOOLER 88
10.6.3 SPECIFYING A RESIDENT SPOOLER 91

10.7 SPOOLER OPERATOR DIRECTIVES 94

10.7.1 HOLD,n 94
10.7.2 RESUME,n 94
10.7.3 REPEAT,n 95
10.7.4 PRINT,n 95
10.7.5 KEEP,n 95
10.7.6 FLUSH,n 95

10.7.7 REMOVE,n 96
10.7.8 LINES,#nxxx 96
10.7.9 READ,#nxxx 96
10.7.10 WRITE,#nxxx 96
10.7.11 INITIAL,#nxxx 96
10.7.12 abc,n 96
10.7.13 STATUS,n and ALL 97

10.8 SPOOLER OPTIONS 98
10.9 SPOOLER CARRIAGE CONTROL 99

CHAPTER 11 MAX IV INPUT SPOOLING SYMBIONT 101

11.1 PHYSICAL INPUT 101

11.1.1 JOB CONTROL CARDS 101
11.1.2 $JOB 101
11.1.3 $RDS 102
11.1.4 $EOJ 102

11.2 IMAGINARY INPUT 102

11.2.1 I/O REQUESTS 102
11.2.2 SPOOLED INPUT - BATCH TASK 102
11.2.3 SPOOLED INPUT - NON-BATCH TASK 103
11.2.4 SPOOLED INPUT EXAMPLES 103

11.3 RDS OPERATOR COMMANDS 103

11.3.1 DISPLAY JOB STATUS 104
11.3.2 REMOVE JOB FROM JOB QUEUE 104
11.3.3 ADD CARD READER(S) 105

11.4 INSTALLING THE INPUT SPOOLER 105

11.4.1 SYSTEM GENERATION 105
11.4.2 CATALOGING RDS 106

11.5 EXISTING JOB CONTROL DECKS 108

APPENDIX A INTERNAL REPRESENTATION OF USASCII CODES 109
APPENDIX B CHARACTER CODES 111
APPENDIX C CARRIAGE CONTROL FUNCTIONS FOR PRINTING DEVICES ... 115
APPENDIX D MODCOMP BINARY CARD FORMAT 117

INDEX 119

LIST OF ILLUSTRATIONS

Figure 2-1. UFT Option Word for TY 11
Figure 2-2. Teletypewriter Handler Device-Dependent Bits 13

Figure 3-1. Scan READ ASCII Mode (Paper Tape Reader) 17
Figure 3-2. Punched Paper Tape Records (Binary Formats) 19
Figure 3-3. Scan READ Binary Mode(Paper Tape Reader) 20

Figure 4-1. Punched Paper Tape Records (ASCII Formats) 28
Figure 4-2. Punched Paper Tape Records (Binary Formats) 29

Figure 5-1. Card Reader Data Paths from Card to Memory 37
Figure 5-2. Packing Raw Columns into Memory 41

Figure 6-1. Card Punch Data Paths from Memory to Card 49

Figure 7-1. Line Printer Handler Device-Dependent Bits 60

Figure 9-1. Plotter Control Information 71

Figure 10-1. Imaginary Printer Device Option Word 87

LIST OF TABLES

Table 2-1. Teletypewriter Operations and Modes 15
Table 2-2. Teletypewriter Errors and Events 16

Table 3-1. Paper Tape Reader Operations and Modes 24
Table 3-2. Paper Tape Reader Errors and Events 25

Table 4-1. Paper Tape Punch Operations and Modes 32
Table 4-2. Paper Tape Punch Errors and Events 33

Table 5-1. Card Reader Operations and Modes 44
Table 5-2. Card Reader Errors and Events 45

Table 6-1. Card Punch Operations and Modes 52
Table 6-2. Card Punch Errors and Events 53

Table 7-1. Line Printer Operations and Modes 61
Table 7-2. Line Printer Errors and Events 62

Table 8-1. Raster Printer/Plotter Operations and Modes 69
Table 8-2. Raster Printer/Plotter Errors and Events 70

Table 9-1. Incremental X-Y Plotter Operations and Modes 75
Table 9-1. Incremental X-Y Plotter Errors and Events 76

CHAPTER 1
INTRODUCTION

This manual contains information about the MAX IV Input/Output handlers that support unit record devices. These devices include the following:

- Console keyboard
- Paper tape reader
- Paper tape punch
- Card reader
- Card punch
- Line printer
- Plotter
- Spooling symbionts

1.1 HANDLER'S RESPONSE

Each handler is explained in a separate chapter. Information about how the handler responds to the following I/O operations is included:

- READ
- WRITE
- REWIND
- BKFILE
- BKRECORD
- AVRECORD
- AVFILE
- WEOF
- HOME

The basic modes of data transfer are:

- Standard ASCII
- Non-Standard ASCII
- Standard Binary
- Non-Standard Binary

Differences, if any, in the handler's response to the four basic modes of data transfer are reported. A chart summarizing the handler's response to each I/O operation in each mode is presented.

After an I/O operation is performed, bits are set in the programmer's UFT status word (Word #0) to indicate the error or event conditions that exist for the physical device to which the file is assigned. A chart indicating the meaning of each bit and the reason(s) for setting appears for each handler.

1.2 DEVICE DEPENDENT FEATURES

Device dependent options can be selected in the programmer's User File Table (UFT) when the programmer is using a particular device. These options enable/disable functions unique to certain devices. The presence of this bit in the UFT causes some devices to perform functions not performed by all devices. The device dependent options that apply to each handler are described in each chapter.

Many of the handlers are designed to function with several types of devices. Some of these devices have peculiar characteristics that must be distinquished by the handler. These special functions or device characteristics are identified to the handler at SYSGEN.

1.3 HANDLER SYSGEN

The SYSGEN statements necessary to include the controller(s) and device(s) in the MAX IV system are described. If special parameters are associated with devices supported by a handler they are fully discussed in the chapter on the handler.

1.4 HOLDUP CONDITIONS

Certain situations prevent the handler from continuing operation. An example of this occurs when the physical device the handler wants to use is under exclusive use by another task. These holdup conditions are reported to the operator for resolution. The format of the error message output to the operator is listed along with the common causes of the error.

1.5 SPOOLING SYMBIONTS

The MAX IV Operating System includes two standard spooling symbionts. The Imaginary Printer Spooling Symbiont, S, handles high-speed buffering of print data between a program and an actual printing device. The Input Spooling Symbiont, RDS, handles high-speed buffering between a program trying to read card data and the actual card reader.

This manual provides information concerning the installation and use of both S and RDS. The SYSGEN statements and cataloging procedures are listed and a detailed description of the operator commands that are available with each is included.

CHAPTER 2
MAX IV HALF-DUPLEX TELETYPEWRITER/CRT HANDLER (TYn)

The logical device TYn combines the keyboard and printer as a single interactive device called a terminal. A terminal can be a conventional teletypewriter (ASR/KSR), alphanumeric CRT, or other entry/display device capable of transmitting ASCII unit records. A terminal (TYn) always operates as a half-duplex device (input/output transmitted on same channel, never simultaneously) when connected to this handler.

The optional paper tape reader (PRn) and punch (TPn) that might belong to a terminal are supported as separate logical devices, since they are often programmed in a unique and independent manner from the keyboard/printer. For more information, refer to the chapters about the paper tape reader handler and the paper tape punch handler.

The teletype handler is a resident MAX IV element consisting of a task level preparation routine and directly connected SERVICE and DATA interrupt routines utilizing unique hardware party-line interrupt vectors. These are permanent resident elements capable of controlling the transfer of data, status, and commands to and from a teletypewriter or similar device. The hardware controller supports only one teletypewriter. However, additional devices may be added with additional controllers if each is assigned a unique group/unit number. A single copy of the handler serves all of them. In the CLASSIC 786X, the teletype controller is part of the Multifunction Controller or the Disc Console Controller.

2.1 STANDARD CONTROLLERS AND DEVICES

The standard teletype controller has the name TT and uses the group/unit #0A. Additional teletypewriter-like devices must be assigned controller names of the form TTx, where x is a unique character distinguishing each device. No Direct Memory Processor (DMP) hardware is required, since the data rate is sufficiently low to use data interrupts for each character transmitted. The standard logical devices for this controller are TY, TP, and PR. These must be assigned to the controller TT at system generation. If a KSR-type device is used, the TP logical device may be dropped. The PR device must be retained if a high-speed paper tape reader exists in the hardware configuration.

This handler is designed to operate using many controllers. These controllers may be connected directly to a terminal device only and not through a data set (modem). The Multifunction and Disc/Console controllers are supported, as well as the Input/Output Interface Subsystem Models 1115 and 1116.

This handler also supports other unbuffered printers such as the GE TERMINET (30 or 120 cps) and the Texas Instruments Silent Terminal (30 cps). These devices have special carriage control options that must be declared in the option word of the logical device table at system generation if hardware for the console

teletype controller cannot support these devices or if these devices are not operated by a teletype controller. CRT devices utilize the option word if a FORM-FEED (FF) byte moves their cursors to the top left corner of their screens. Other teletypewriter devices have form-feed simulated with multiple CR/LF sequences.

NOTE: Special hardware controllers that multiplex as many as 256 terminal devices in FULL-DUPLEX fashion are available from MODCOMP as "communications extensions", but they are not supported by this handler. For information concerning these handlers refer to the MAX IV COMMUNICATIONS HANDLERS, System Guide Manual.

2.2 I/O OPERATIONS AND MODES

2.2.1 STANDARD ASCII READ FROM KEYBOARD (TYn)

On Standard ASCII READ operations, each input character is automatically returned from the controller to the printer (echo). On a READ operation, the unit record transmitted is a line of characters usually delineated by a special ASCII byte called the terminating character. For the Standard ASCII mode, this may be either a trailing CARRIAGE RETURN (CR) or a trailing NUL (gap) character. Trailing characters are control characters received after at least one valid data character.

The default record length for TYn devices is 74 bytes. The 74 byte record may include 72 data characters, a CARRIAGE RETURN, and a LINEFEED character, that is, 72 + CR/LF = 74. Other carriage widths from 4 to 510 bytes can be specified as a unit record limit for a TY device at SYSGEN.

The carriage return and null control characters are ignored if they are leading. Leading characters are control characters received before a valid data character is received by the handler.

The control character LINEFEED (LF) is "filtered out" on Standard ASCII operations and is never put into the user's data buffer whether leading, trailing, or intermediate. Leader skipping may be defeated by option Bit 1 in the User's File Table (UFT) if they are required characters. Refer to the UFT OPTIONS Section in this chapter.

2.2.1.1 Prefix Message

Before a READ operation is initiated to a keyboard (TYn), a short prefix message consisting of only a carriage return and a line feed is transmitted to the printer. This prefix serves to:

- Inform the operator that an input is queued.
- Position the carriage on a unit record boundary (start of next line) in case it was left in an indeterminate position by the previous print or input operation.

This carriage positioning prefix may be overridden by the Device-Dependent Option bit in the UFT if the programmer wishes to proceed with input on the same line as a previous print operation. Of course, the programmer must deal with an interactive device in such a case rather than the general input device that the STANDARD ASCII mode attempts to simulate.

2.2.1.2 Special Purpose Characters

Error correction of operator type-in is local to the handler in this mode and requires that certain ASCII control characters be reserved for special purposes:

- Delete character
- Delete record
- Suppress trailing blanks
- Detect intermixed Binary and ASCII records
- Detect OC task or TMP request
- Detect end-of-file
- Detect null record

The BACKSPACE (CONTROL "H") character is not passed as a data byte, but causes the previously entered data byte and the BACKSPACE (BS) itself to be deleted from the unit record. If successive BS bytes are entered, they delete an equal number of previously typed characters. If more BS bytes are entered than there are previous characters, the extra BS bytes are ignored.

The DELETE (DEL) (sometimes called RUBOUT) character cancels all previously entered bytes in the unit record and restarts the READ operation, also without program assistance. Thus, the typist, not the programmer, must be concerned with this type of error.

All trailing blanks (SPACE) are suppressed and a NULL word (two NUL bytes on a word boundary) marks the end of useful information in the memory buffer as long as there is still room in the buffer to store these extra bytes. A user may define such a character sequence (DFC Ø) at the end of the input buffer if the possibility exists that the unit record will fill the entire buffer. This trailing blank suppression feature, though rarely necessary when the keyboard is being operated interactively, ensures that input records received from keyboard, card, and paper tape will look the same to a programmer.

The Standard ASCII mode is capable of detecting the presence of a Standard Binary formatted record should it encounter one. Such records are identified by a byte with a binary value 'ØØØØØØ11 or 'ØØØØØ111, which corresponds to the ASCII bytes "end-of-text" (ETX) and "bell" (BEL). These data bytes should never be used to start a Standard ASCII record. This record mode identification feature is used to permit the reading of STANDARD records that may be intermixed Binary and ASCII records. This is discussed in more detail in the chapter describing the PRn device.

If a CONTROL "A" is detected in the Standard ASCII READ from the keyboard, a special routine is called to check for the Operator Communications Task (OC) or the Terminal Monitor Program (TMP) using this terminal. If either task is using the terminal, it will be activated. If neither task is using the terminal, CONTROL "A" will be treated as a data character and placed in the buffer. CONTROL "A" detect only works while the terminal is not being written to or while reading a Standard ASCII record from the keyboard.

If BREAK (on controllers capable of supporting BREAK) is detected by the controller, a routine is called to check for either the Operator Communications Task (OC) or the Terminal Monitor Program (TMP) using this terminal. If either task is using the terminal, it will be activated. If neither task is using the terminal, BREAK is treated as a NUL.

After a READ operation is terminated, the first word (Bytes Ø and 1) is examined by the handler. If it contains the ASCII bytes "$$", the end-of-file (EOF) indicator is turned on in the programmer's UFT status word.

A character sequence: ....LF/CR....

is treated as a "null" or "blank" record and a NUL marker is put in Word Ø of the programmer's data buffer.

2.2.2 NON-STANDARD ASCII READ FROM KEYBOARD (TYn)

A READ operation to TYn in the Non-Standard ASCII mode is not as "automatic" as the Standard ASCII mode. It requires that the programmer be aware that the input file is assigned to a teletype keyboard/printer specifically. The main purpose of the mode is to allow for device-dependent programming when the automatic editing and control character interpretation of the Standard ASCII mode are restrictive to the programmer's application. This mode requires more work for the programmer. This type of program cannot have its file indiscriminately assigned to any input device.

This mode does not output the CR/LF prefix to the printer before a READ operation is started. None of the local editing bytes (BS or DEL) are interpreted in the Non-Standard ASCII mode nor are trailing blanks suppressed or line feeds ignored. The size of variable-length unit records may be determined from the byte count returned to the UFT or by the position of a particular terminating character in the buffer. CONTROL "A" is not detected in this mode.

Null (NUL) bytes are bypassed as leader and a single NUL byte terminates the Non-Standard ASCII unit record. However, any other ASCII byte can be selected by the programmer as the record terminating byte. This byte marks the end of the record in the buffer. Refer to the TERMINATOR BYTE section in this chapter. The leader-skipping function can be defeated if option Bit 1 (DDO) is set in the UFT. Refer to the UFT OPTIONS section in this chapter.

The end-of-file event is never detected in this mode, even if the first two data bytes of the unit record are "$$".

2.2.3 BINARY READ FROM KEYBOARD (TYn)

Teletypewriter keyboards are assumed to interface directly with human operators; thus, binary read operations are not defined. These modes are interpreted by the paper tape devices that interface to terminals. Binary modes may be used by special terminal handlers that have both graphic and character modes.

2.2.4 STANDARD ASCII WRITE TO PRINTER (TYn)

The teletypewriter printer transmits one unit record (a line) for each WRITE operation. In the Standard ASCII mode, the text transmitted by the programmer must have a carriage control byte as the first byte (Ø) in the buffer, which is not printed. This byte indicates to the handler which printer carriage positioning function is to be performed before the remainder of the buffer is printed.

Since carriage control is performed before printout and as an integral part of the WRITE operation, the operation frees the programmer from remembering from one line to the next where the carriage is positioned. The carriage control character also facilitates transmission of ultimate carriage positioning information when the printed data is diverted to secondary storage by the system (spooling) without the programmer's knowledge. A special task (symbiont) can use the carriage control information when it subsequently prints the diverted print data on a physical printer.

The next byte (1) following the carriage control byte may be the start of printed text. However, if this is inconvenient, text may start on the next word (byte 2) boundary if byte 1 is a "start-of-text" (STX) byte. This deferred text start is often convenient where WRITE operations consist of data recently received by a READ command from another device. Since the READ operation always stores data in memory starting on a word boundary, the carriage control byte and STX character can simply be put in a word that precedes the READ buffer when it is necessary to print. The print and input buffers are thus coincident (though out of phase by one word), and moving of the data is not necessary when the carriage control information is added.

```
PRBUF     DFC       #2Ø02        Print carriage control
INBUF     RES        41,Ø        Input buffer
```

The carriage control bytes are defined in Appendix C. Most teletypewriters have no hardware carriage control capability other than a form-feed control byte; thus, the carriage control function, performed when the most significant bit of the carriage control byte is set, is to transmit "n" line feeds (LF) where "n" is the value of the remaining 7 bits.

Since USASCII is a 7-bit code, all characters transmitted to the printer have their most significant bits removed (set = Ø) in each 8-bit byte when taken from memory. Text bytes are transmitted until a null (NUL) byte marker is found in the user's buffer or until the device's record size or the buffer's specified size is exceeded. Thus, a facility for both variable- and fixed-length record sizes is provided, specified either by call argument or by the contents of the data buffer. If the user's buffer is larger than the device's record size, the additional characters are not printed. The use of the NUL byte marker is convenient as a terminator when the program generating the messages is not related to the program calling the WRITE operation. The latter can always be made to specify a buffer that is the maximum device size while the former can communicate to the handler the size of the useful text line by imbedded control bytes in the buffer.

Except for the conventions described above, all other text bytes and control bytes are transmitted to the printer. Even additional carriage control characters may be imbedded in text (for example, CR,LF,FF), but this convention should not be used if the program might need to have its file assigned to other printing devices that may not control carriage motion with data bytes (for example, line printer). The use of the carriage control bytes and other imbedded non-printing bytes in the buffer must be accounted for in the device's record size and buffer size. All printing devices generally have their record sizes specified two bytes larger than actual carriage width at system generation to account for carriage control bytes. A 72-column printer has 74 bytes specified; a 132-column printer has 134 bytes specified; and so on.

2.2.5 NON-STANDARD ASCII WRITE TO PRINTER (TYn)

In the Non-Standard ASCII mode, no carriage control character is interpreted on WRITE operations; thus, all carriage control must be performed by imbedded control bytes in the data text buffer. Therefore, the programmer must know that the file is assigned to TYn before using this mode. All 8-bit data bytes in memory are converted into 7-bit USASCII bytes before being transmitted (most significant bit of the 8-bit memory byte is removed).

This mode transmits all characters in the programmer's data buffer until one of the following occurs:

- The device's record size is exceeded
- The specified buffer size is exceeded
- A byte in the buffer matches the byte stored in the programmer's UFT option word.

Thus, variable-length records, whose size is determined by imbedded data bytes, can be transmitted in this mode also. If bit 8 of the option word is set, however, the feature is defeated and only fixed-length records can be transmitted. This mode normally terminates upon finding a NUL byte in the data buffer if the programmer takes no action to set the desired byte in the UFT or to defeat the feature. The terminating byte is not transmitted.

2.2.6 FILE/RECORD OPERATIONS

2.2.6.1 ADVANCE RECORD (TYn)

The ADVANCE RECORD operation causes the printer carriage to be upspaced regardless of data mode. A carriage return (CR) and a line-feed (LF) byte are transmitted. No function is performed for the keyboard for this operation.

2.2.6.2 ADVANCE FILE (TYn)

The ADVANCE FILE operation causes the printer carriage to form feed (FF) regardless of mode. Some printers position the carriage to the top of the next page. Printers without form feed hardware simulate form feed by transmission of one carriage return (CR) byte followed by four line feed (LF) bytes. Bit 4 in word 3 of the Logical Device Table (LDT) and bit 4 of word 4 of the Transport Control Table (TCT) must be set at system generation if the form feed capability exists in the device.

2.2.6.3 REWIND and HOME (TYn)

The REWIND and HOME operations cause form feed of the printer just as ADVANCE FILE. The programmer is encouraged to perform either of these operations at the beginning of the program if the position of the carriage is unknown. This can also be accomplished by printing the first line at top of form. The latter method is required if a "spooled" device, which buffers printing data to secondary storage, is used.

2.2.6.4 BACKSPACE RECORD/BACKSPACE FILE (TYn)

These BACKSPACE operations do not transmit data to the teletypewriter but act as null operations. The EOF and BOM event bits are set in the programmer's UFT.

2.2.6.5 WRITE END-OF-FILE MARK (TYn)

This WRITE operation causes a "$$" message to be printed with a carriage return (CR) and two line feeds (LF) preceding it. The EOF event bit is set in the programmer's UFT.

2.2.6.6 Summary Tables

Teletype Handler summary tables appear at the end of this chapter. Table 2-1 contains information about the handlers response to I/O operations in the four modes of data transfer. Table 2-2 lists the error and event bits in the UFT status word (UFTSTA). Conditions causing the error or event are also listed.

2.3 SPECIAL QUEUING RULES

The console teletype, is a central communications device for many tasks and is used to report many system error messages. For these reasons it has special I/O queuing rules applied to it so that it will always be able to serve and inform the system operator without delay. These rules may optionally apply to other terminal devices that share this handler if a system generation option is specified.

The basic difference is that the currently initiated operation in the queue (the first node) may be terminated at "mid-operation" and requeued later if more important operations are required. This is avoided on non-interactive devices, since partially completed data records should be avoided in most cases.

Each terminal device has a status bit in its Logical Device Table that, when reset, causes all output (WRITE) operations to have priority over input (READ) operations. This is the "normal" way MAX IV lets such a terminal operate, on the theory that messages from any number of tasks and system services must always be allowed to occur since they keep the operator informed. Likewise, output (WRITE) from higher priority tasks may even interrupt outputs from lower priority tasks with the interrupted messages being retyped from the beginning when higher priority messages subside.

This condition is acceptable in most appplications most of the time, except that it is always possible for an operator to be so overwhelmed with messages that he/she is unable to act; that is, the operator is not able to enter a directive to correct the conditions causing the flood of messages if the state of output messages (WRITE) having priority over input messages (READ) continues. In this case, the special status bit is set from the time the Console Interrupt switch is depressed until one successful input directive is received from the operator and acted upon by the system or until the "watchdog" timer expires and shuts down the OC task's read. Thus, the opposite queuing rule -- READ has priority over WRITE -- is imposed during this time window so that an operator can be assured of taking action.

All messages (WRITE) are held in the queue as long as an input (READ) is queued. If more than one input directive must be typed by the operator in order to take control, he/she may have to depress Console Interrupt before each directive. This special status bit is currently set only for the console teletype in MAX IV, but the programs that operate other remote terminal devices can set or reset this bit in their device tables to effect the same queuing rules. The default rule is output over input. These functions affect only the TYn device, leaving operations queued on the PRn and TPn devices undisturbed. The operator is allowed to communicate between PRn and TPn unit records, however.

Additional queuing rules are imposed on the console teletypewriter ONLY. A special bit is set whenever the Console Interrupt has been depressed four times without a successful input directive being queued. This bit, called the EMERGENCY QUEUING bit, causes all operations queued by the highest priority tasks to be initiated immediately upon request, regardless of what operations are already

in the queue. This causes operations from auxiliary teletypewriter devices (PRn and TPn) to also be affected (terminated), as well as the TYn device. The operator is in a position of judgment and must make the decision whether the transmission of PRn or TPn records must be sacrificed for the sake of gaining immediate control of the system. It will also be the operator's duty to abort tasks using these secondary devices, if their premature termination would cause unrecoverable data errors.

2.4 UFT OPTIONS

The user indicates the data format mode and various options selected by setting bits in the UFT Option word. This word is illustrated in Figure 2-1 and described in the following sections.

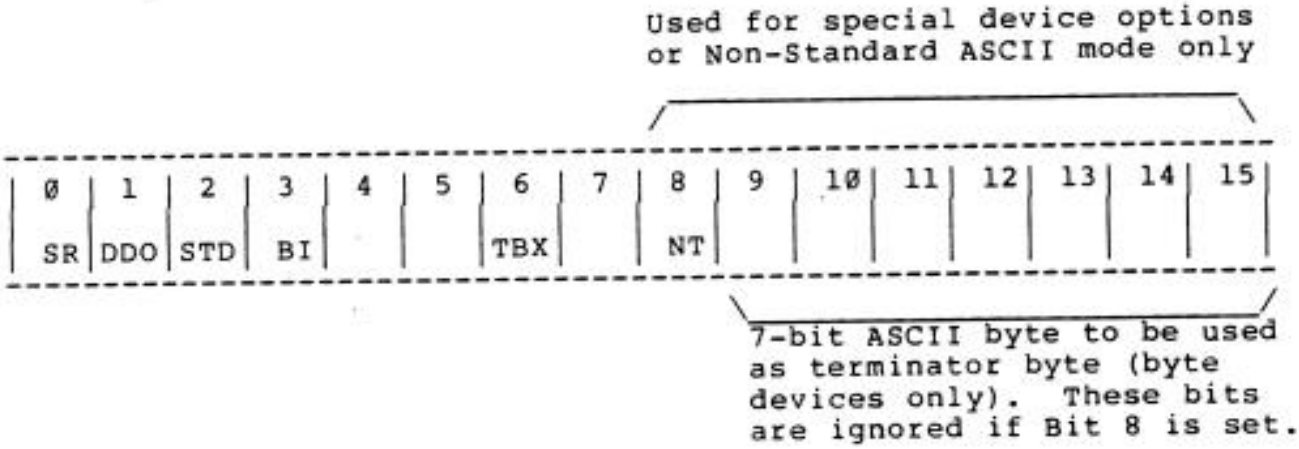

Used for special device options or Non-Standard ASCII mode only
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
SR DDO STD BI TBX NT
7-bit ASCII byte to be used as terminator byte (byte devices only). These bits are ignored if Bit 8 is set.

Figure 2-1. UFT Option Word for TY

Bit	Name	Description
0	SR	System Recovery - automatic recovery if errors detected and will notify operator instead of returning to calling task.
1	DDO	Device-Dependent Option - device functions unique to certain devices.
2	STD	Standard Format. The Non-Standard data format mode is selected if this bit = 0.
3	BI	Binary Mode. The ASCII data format mode is selected if this bit = 0.
6	TBX	Inhibit data editing functions (trailing blank, suppression, et cetera).
8	NT	No terminator byte for Non-Standard ASCII in Bits 9-15.

2.4.1 DEVICE-DEPENDENT OPTION

The Device-Dependent option is selected by setting Bit 1 (DDO) of the UFT option word. The presence of this bit causes some devices to perform functions not performed by all devices. Successful use of this feature requires that the user's logical file be assigned to a particular device with the ability to respond as expected to the DDO.

The Keyboard (TYn) handler responds to the Device Dependent Option in the following manner. When Bit 1 of UFTOPT is set, the normal carriage positioning prefix (CR/LF) is not output before a READ operation, thus allowing input responses on the same line as printed messages.

2.4.2 INHIBIT DATA EDITING

A special control bit modifies the device handler's response during a READ at the end of a unit record. This bit (Bit 6 of the UFT option word), when set, defeats end-of-line detection, trailing blank suppression, and the storing of the NUL byte marker at the end of useful information. In this special mode, the programmer can determine the exact number of bytes read, and the end of useful information in the buffer, only by a byte count stored in the UFT.

2.4.3 TERMINATOR BYTE

In Non-Standard ASCII mode, the programmer can select any ASCII byte as the record terminating byte. The selected terminating byte must be in bits 9 through 15 of the UFT option word when the operation is queued. The handler will terminate the Non-Standard ASCII record when the selected terminator byte is read.

If Bit 8 of the UFT option word is set, the premature termination feature is defeated.

2.5 HOLDUP CONDITIONS REPORTED TO OPERATOR

A message reporting a holdup condition is output to the operator in the following format:

```
!TYn?/tttttt/oooooo
```

tttttt - is the calling task's name

oooooo - is the current task/overlay name

The message reports the following conditions:

- Device does not exist
- Device under exclusive use by another task
- Device already offline
- Centronics printer offline switch set

2.6 SYSTEM GENERATION

2.6.1 STANDARD PARAMETERS

The following statements are required in a system generation source file if this device is to be included in a system. Detailed descriptions of these statements and the rules of order in which they must be entered are included in the MAX IV SYSGEN, System Guide Manual.

```
CONTROLLER          TTx,#A,,UNTIMED
    .
    .
    .
DEVICE              TYn,TTx,[carriagewidth],[option]
```

where "x" is a blank for only the first controller, and where "n" is a blank for only the first device, and where "n" and "x" are 1, 2, 3 ... for additional devices and controllers. The optional "carriagewidth" defaults to 74 unless specified. The "option" parameter is described as follows, and must be expressed as a hexadecimal number.

2.6.2 DEVICE OPTION PARAMETER

The MAX IV teletypewriter handler is designed to function with different types of devices. However, some of these devices have peculiar characteristics that must be distinguished by the handler. The following bits may be set in the TCT and LDT option words at system generation to identify the device and its special functions or characteristics (these bits are coded in the Hexadecimal format at on parameter 4 of the DEVICE statement). For example, if parameter 4 of the DEVICE statement was specified as #1824, that would indicate the following bits set: N08 and CFF and PTR and TRM. An explanation of each of these options follows:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	N36	N08	CFF	NEK	0	0	0	0	PTR	0	0	TRM	0	0

Figure 2-2. Teletypewriter Handler Device-Dependent Bits

Bit	Meaning If Set (=1)
N36	Device is a GE TERMINET 1200 that requires 36 null characters be output to provide time for carriage control operations. This is a 1200 baud device.
N08	Device is a GE TERMINET 300 or a TI Silent 700 that requires 8 null characters be output to provide time for carriage control operations. These are 300 baud devices.
CFF	Device has hardware form-feed capability. Handler may issue a special form-feed character (FF) to accomplish form feed. When this option is not specified, a series of carriage return (CR) and four line feeds (LF) simulate form-feed operations.
NEK	Device is a special CRT; do not echo keyboard input to screen. This mode can be supported only if the CRT is connected to a SACI board (part of the IOIS) or the standard teletype controller has been modified so that it does not support the paper tape reader's two character buffer.
PTR	Must be set (=1) to indicate that the device can print.
TRM	Set if keyboard used (for example, device is complete terminal) or reset if device is a print-only device.

TELETYPEWRITERS/ALPHANUMERIC
CRT's (TYn), Centronics Printers (TYC)

74 ≤ (RECORD SIZE) ≤ 134 BYTES

	<----------ASCII MODES---------->		<----------BINARY MODES---------->	
	STANDARD	NON-STANDARD	STANDARD	NON-STANDARD
0 READ	INPUT ONE ENTRY LINE FROM KEY-BOARD. TERMINA-TOR = CR OR NUL *OUTPUT PREFIX (CR,LF) BEFORE INPUT	INPUT ONE ENTRY LINE FROM KEY-BOARD (USER SPECIFIED TERMI-NATOR) -NO PREFIX- TO PRINTER	UNIMPLEMENTED	UNIMPLEMENTED
1 WRITE	OUTPUT ONE LINE TO PRINTER; OUT-PUT PREFIX AS PER CARRIAGE CONTROL BYTE INTERPRETATION	OUTPUT ONE LINE TO PRINTER -NO PREFIX- TO PRINTER	UNIMPLEMENTED	UNIMPLEMENTED
2 REWIND	OUTPUT SIMULATED (CR,4LF) OR TRUE (FF) FORM-FEED TO PRINTER.+	<-------SAME	UNIMPLEMENTED	UNIMPLEMENTED
3 BKFILE	null+	<-------SAME	UNIMPLEMENTED	UNIMPLEMENTED
4 BKRECORD	null+	<-------SAME	UNIMPLEMENTED	UNIMPLEMENTED
5 AVRECORD	ADVANCE CARRIAGE OF PRINTER ONE LINE (CR,LF)	<-------SAME	UNIMPLEMENTED	UNIMPLEMENTED
6 AVFILE	OUTPUT SIMULATED (CR,4LF) OR TRUE (FF) FORM-FEED TO PRINTER	<-------SAME	UNIMPLEMENTED	UNIMPLEMENTED
7 WEOF	OUTPUT FILE-MARK ($$) RECORD TO PRINTER WITH CR,2LF) LEADER PREFIXED, SET EOF EVENT BIT IN UFT.	<-------SAME	UNIMPLEMENTED	UNIMPLEMENTED
8 HOME	OUTPUT SIMULATED (CR,4LF) OR TRUE (FF) FORM-FEED TO PRINTER+	<-------SAME	UNIMPLEMENTED	UNIMPLEMENTED

*This function is defeated if Bit 1 of the UFT option word set.
+This operation sets the EOF and BOM events in the UFT status word.

Table 2-1. Teletypewriter Operations and Modes

PROGRAMMER'S ERROR BIT IN UFT WORD UFTSTA	OPERATOR'S REASON CODE	REASON FOR ERROR IF SET
1	OVF OVERFLOW/ UNDERFLOW	-NONE- DEVICE IS ASYNCHRONOUS
2	PAR PARITY OR DATA CHECK	-NONE-
3	INO INOPERABLE	-DEVICE DOES NOT EXIST -POWER HAS FAILED IN THE CONTROLLER -A TIMED DEVICE HAS TIMED OUT
4	MPE MEMORY PARITY OR IOP ERROR	-AN IOP PARAMETER ERROR HAS OCCURRED (I.E., BUFFER ADDRESS IS OUTSIDE OF TASK'S ADDRESSING RANGE.) SYSTEM RECOVERY CAUSES TASK TO ABORT.
5	LOK WRITE LOCK	-NONE-
6	CIG CHARACTER IN GAP	-NONE-
7	OTH OTHER	-NONE-
8	SBV STANDARD BINARY VIOLATION OR DMP ERROR	-STANDARD BINARY OUTPUT RECORD IS LARGER THAN OUTPUT BUFFER. (SYSTEM RECOVERY CAUSES THE TASK TO ABORT.)
PROGRAMMER'S EVENT BIT IN UFT WORD UFTSTA	**REASON CODE**	**REASON FOR EVENT IF SET**
9	EOM END-OF- MEDIA	-PAPER LOW (CENTRONICS)
10	EOF END-OF- FILE	-END-OF-FORM (CENTRONICS) -STANDARD ASCII READ ENCOUNTERED A "$$" -A NULL OPERATION WAS REQUESTED
11	BOM BEGINNING- OF-MEDIA	-A NULL OPERATION WAS REQUESTED

Table 2-2. Teletypewriter Errors and Events

CHAPTER 3
MAX IV PAPER TAPE READER HANDLER (PRn)

The paper tape reader is controlled in a manner similar to the basic teletype handler. The handler implements both ASCII and Binary modes. The high-speed paper tape reader and the paper tape reader that is part of an ASR teletypewriter appear identical to the handler. The slow reader is used if the high-speed reader is missing or its power is off. The READ operation from the paper tape reader (PRn) operates in a manner similar to TYn.

3.1 I/O OPERATIONS AND MODES

3.1.1 ASCII READ FROM PAPER TAPE READER (PRn)

The carriage positioning prefix output for TYn device READ operations are not transmitted and no received characters are echoed to the printer; however, all leader-skipping and error-correction functions operate in the same manner as TYn for both Standard and Non-Standard ASCII modes.

The paper tape reader has an indefinite record size (but not exceeding 32,768 bytes). If a paper tape record is encountered that fills the programmer's buffer before the terminating character is detected, no more data is transmitted, but the remaining characters in the paper tape record are bypassed until a terminating character is found. This function is useful in scanning for "keys" near the beginning of header records where the remainder of the record need not be transmitted into memory. Refer to Figure 3-1.

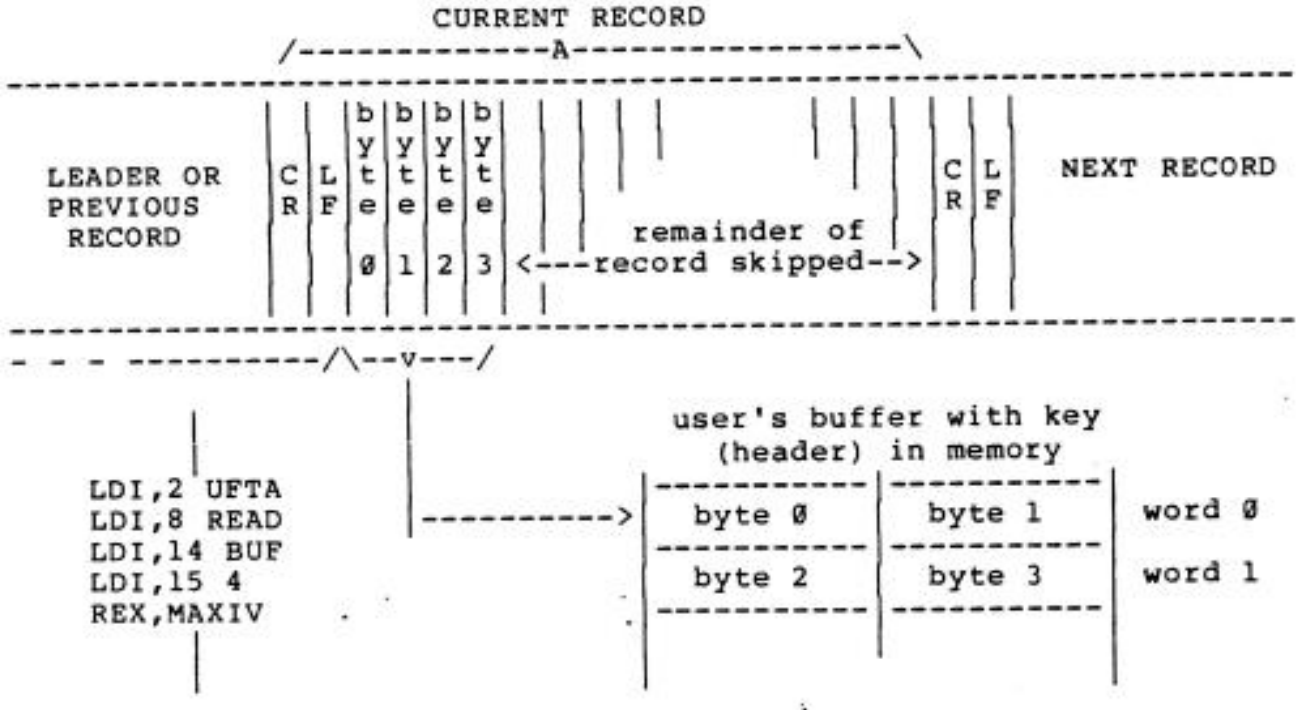


Figure 3-1. Scan READ ASCII Mode (Paper Tape Reader)

3.1.1.1 Hardware Buffer

The paper tape reader always retains the two characters that follow a terminating character. This is due to the inability of the mechanical hardware device to stop between characters. These bytes are retained in a hardware buffer register in the controller and are presented as the first two data characters of the next unit record.

When a new program accesses this reader, it needs to reset this hardware buffer so that these "old" bytes are not transmitted, since they may exist from a previous user's tape that has been removed. A REWIND or HOME operation causes a status bit to be set within the physical device table so that the next READ to PRn causes the hardware buffer to be reset. When READ operations span more than one physical paper tape, the programmer should provide for trailer following the last record of the previous tape or should REWIND or HOME the reader when the new tape is placed in the reader.

3.1.1.2 Automatic Mode Switch

For Standard ASCII mode, the handler automatically switches to Standard Binary mode if a Standard Binary record is detected. This permits intermixed ASCII and Binary records to exist on the same medium, a feature useful to some special purpose programs. This also allows the file mark ($$) record to be detected in both Binary and ASCII data modes.

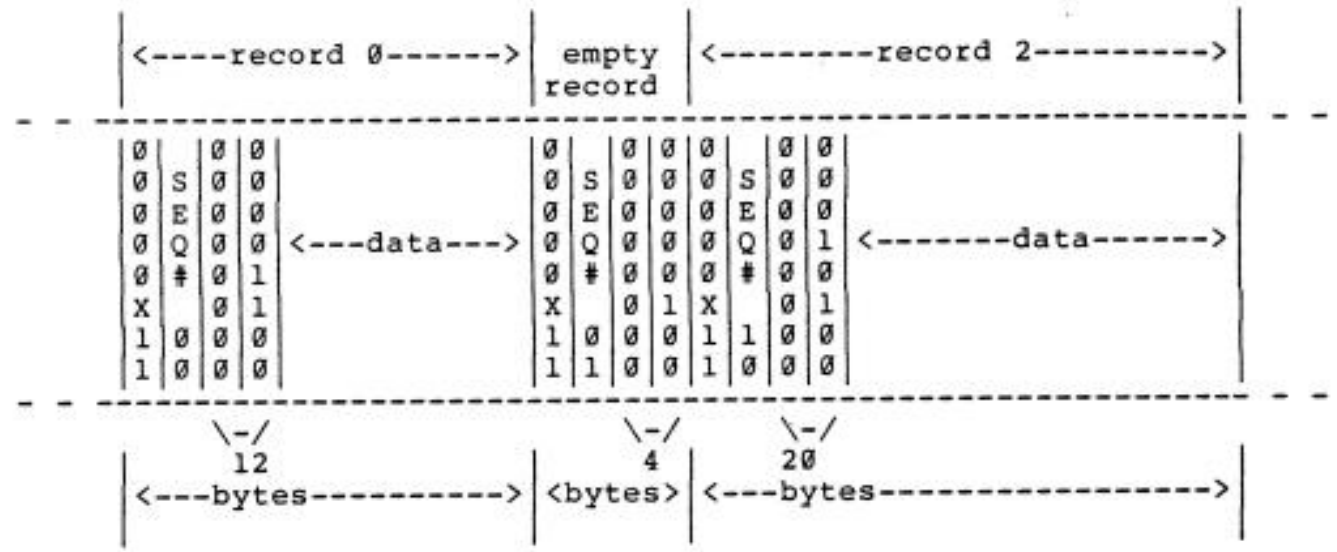
3.1.2 BINARY READ FROM PAPER TAPE READER (PRn)

The binary modes must be used when full 8-bit data bytes are to be transmitted from the paper tape reader. The Standard Binary mode requires a strict format for the first four bytes of the record. These four data bytes uniquely specify the type of record (Standard Binary) and its total length in bytes. This format is shown in Figure 3-2.

If the size of the Standard Binary record is larger than the programmer's buffer, the portion of the record that cannot be contained in the buffer is skipped. This allows for record scanning when keys are being sought that reside near the beginning of a record and the entire record need not be transmitted into memory. Refer to Figure 3-3.

The Non-Standard Binary mode has no strict format, but unit record sizes must be known to the programmer since only fixed-block length records (the size of the specified buffer) can be read.

Standard Binary Records - variable length



NOTE: X may be = 0 or = 1 in byte 0.
Leader may separate each record.

Non-Standard Binary Records - fixed length

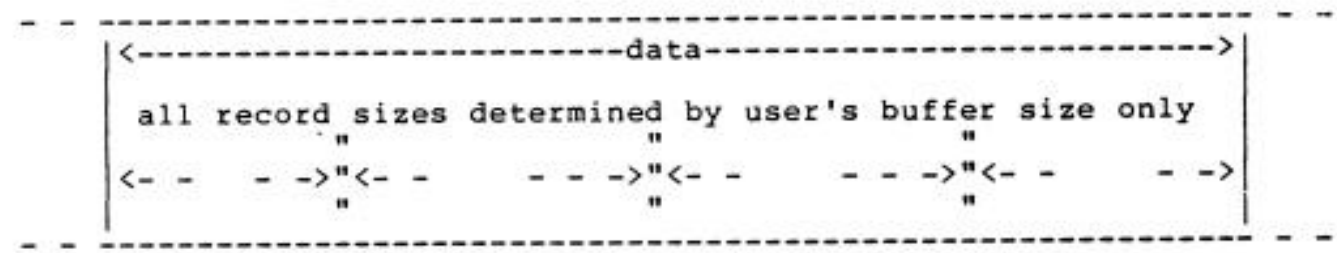


Figure 3-2. Punched Paper Tape Records (Binary Formats)

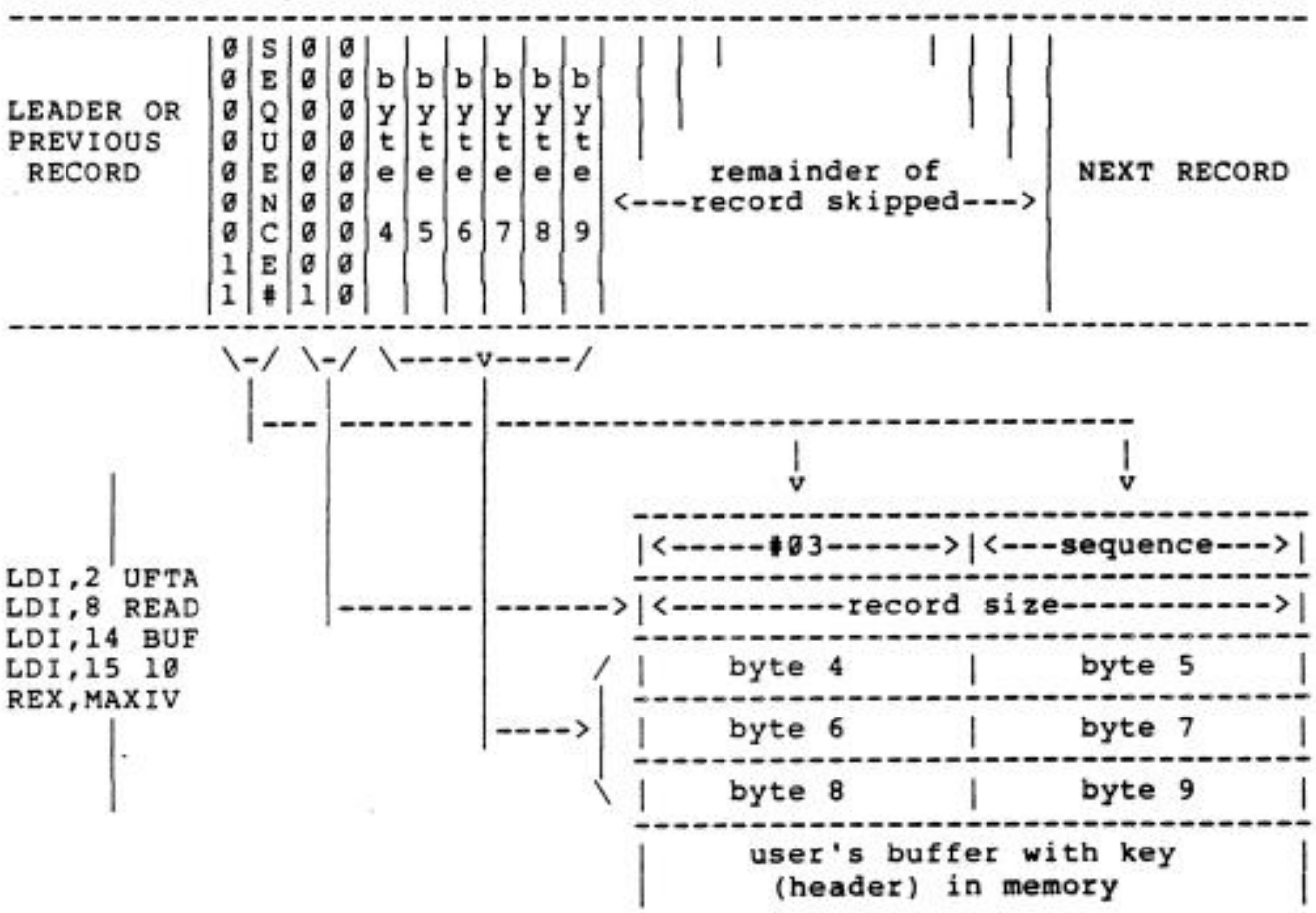


Figure 3-3. Scan READ Binary Mode (Paper Tape Reader)

3.1.2.1 Skip Leader

Both modes skip leader (NUL bytes = '00000000) on each unit record. This feature can be defeated if byte Ø of a Non-Standard Binary record might contain a NUL byte. Refer to UFT OPTIONS.

When NUL bytes might start a record, it is suggested that the first record (dummy if necessary) NOT be allowed to begin with a NUL byte. This allows the leader-skipping function to operate when the tape is mounted. The programmer can defeat the leader-skipping feature after this first record is read. All subsequent records could then contain a NUL byte as the first byte of a record. Without such a convention, the tape must be accurately positioned to data Byte Ø of the first record.

3.1.2.2 Standard File-mark Record

If a STANDARD file-mark record (an ASCII "$$" record) is encountered, the EOF event bit is set in the programer's UFT.

3.1.3 ADVANCE RECORD FROM PAPER TAPE READER (PRn)

The ADVANCE operations have meaning only in the Standard ASCII and Standard Binary modes. A unit record as defined for the particular mode is read without any data being transmitted to a programmer's buffer. The record is simply skipped although it is examined to determine if it is a file mark ($$). If a Standard ASCII record is encountered while the programmer has requested a Standard Binary record or vice versa, the handler switches to the appropriate mode rules and successfully completes the reading of the record in the new mode. This condition is not reported to the operator or the programmer as an error but as a special event (bit 8 of UFT status word) that the programmer may choose to ignore or process. This feature allows intermixed Standard ASCII and Binary records to be processed consistently. It also allows the file mark format to be the same for both modes. If a file mark is detected, the EOF event is set in the programmer's UFT.

3.1.4 ADVANCE FILE FROM PAPER TAPE READER (PRn)

The ADVANCE FILE operation is identical to the ADVANCE RECORD operation except that many records are skipped if necessary until one with a file mark is detected. This operation is requeued after each unit record is detected so that an operation of such indefinite length will not hold back higher priority requests.

3.1.5 FILE/RECORD OPERATIONS

3.1.5.1 REWIND and HOME TO PAPER TAPE READER (PRn)

The REWIND and HOME operations are null but cause the next READ to be initiated with the paper tape reader's hardware character buffers cleared. The REWIND and HOME operations set the EOF and BOM event bits in the programmer's UFT.

Whenever tapes are changed or moved in the reader, the REWIND or HOME function should be performed. It is good practice for a programmer at the beginning of a program, to HOME all files that will be used. A REWIND enables the buffer clear but has the additional function of positioning the logical file to its initial or zero point. The HOME does not reposition the logical file. With the MAX IV I/O concept of supporting the use of substitute devices, the programmer should keep in mind that some substitute devices will reposition the media in response to a REWIND.

3.1.5.2 BACKSPACE RECORD/BACKSPACE FILE TO PAPER TAPE (PRn)

The BACKSPACE operations are not implemented because the paper tape reader does not have reverse operations. They cause the hardware buffer to be cleared on the next READ or ADVANCE as in HOME and REWIND. The EOF and BOM event bits are set in the programmer's UFT.

3.1.5.3 WRITE FROM PAPER TAPE READER (PRn)

This operation is not implemented in any mode; it performs as a null operation. The EOF and BOM event bits are set in the programmer's UFT.

The paper tape reader controller must be operable, however, when such an operation is attempted. When it is desired to punch paper tapes, the logical device TPn (or PPn) must be addressed with a WRITE operation.

3.1.5.4 Summary Tables

Paper Tape Reader Handler summary tables appear at the end of this chapter. Table 3-1 contains information about the handlers response to I/O operations in the four modes of data transfer. Table 3-2 lists the error and event bits in the UFT status word (UFTSTA). Conditions causing the error or event are also listed.

3.2 UFT OPTIONS

The user indicates the data format mode and various options selected by setting bits in the UFT Option word.

3.2.1 DEVICE-DEPENDENT OPTION

The Device-Dependent option is selected by setting Bit 1 (DDO) of the UFT option word. The presence of this bit causes some devices to perform functions not performed by all devices. Successful use of this feature requires that the user's logical file be assigned to a particular device with the ability to respond as expected to the DDO.

The paper tape reader responds to the Device Dependent Option in the following manner. When Bit 1 of UFTOPT is set, the normal function of ignoring leader (NUL for non-standard modes, NUL, CR, and LF for standard modes) is disabled during READ operations. This special read mode may be useful to applications where it is desired to read finite blocks of paper tape that were written in record lengths too large for convenient buffering. Such tapes must be positioned manually to the first character to be read, or Bit 1 in the option word must be set AFTER the first READ command which would normally skip the initial leader.

3.3 HOLDUP CONDITIONS REPORTED TO OPERATOR

A message reporting a holdup condition is output to the operator in the following format:

```
!PPn?/tttttt/oooooo
```

tttttt - is the calling task's name

oooooo - is the current task overlay name

This message reports the following conditions:

- Device under exclusive use by another task
- Device already offline

3.4 SYSTEM GENERATION

The following statements are required in a system generation source file if this device is to be included in a system. Detailed descriptions of these statements and the rules of order in which they must be entered are included in the MAX IV SYSGEN, System Guide Manual.

```
CONTROLLER              TTx,#A,,TIMED
    .
    .
    .
DEVICE                  PRn,TTx
```

Note that the controller TT is the same controller used for teletype (TYn). This controller card should only be entered once for the standard operator's teletypewriter with paper tape reader (fast or slow).

RECORD SIZE ≤ BYTES

	<----------ASCII MODES--------->		<----------BINARY MODES---------->	
	STANDARD	NON-STANDARD	STANDARD	NON-STANDARD
0 READ	INPUT ONE STD. ASCII(OR BINARY) RECORD. TERMINATOR = CR OR NUL *SKIP LEADER OF (NUL...CR...LF) BEFORE EACH RECORD	INPUT ONE USER-DEFINED RECORD. TERMINATOR = USER DEFINED *SKIP LEADER OF (NUL...) BEFORE EACH RECORD	INPUT ONE STD. BINARY (OR ASCII) RECORD. TERMINATOR = COUNT IN BYTES 2, 3 OF RECORD. *SKIP LEADER OF (NUL...CR...LF) BEFORE EACH RECORD	INPUT ONE BLOCK. RECORD = SIZE OF USER'S BUFFER. *SKIP LEADER OF (NUL...) BEFORE EACH RECORD
1 WRITE	null+	<------SAME	<---------SAME	<-------SAME
2 REWIND	null+#	<------SAME	<---------SAME	<-------SAME
3 BKFILE	null+#	<------SAME	<---------SAME	<-------SAME
4 BKRECORD	null+##	<------SAME	<---------SAME	<-------SAME
5 AVRECORD	SKIP ONE STANDARD ASCII (OR BINARY) RECORD, TERMINATOR = CR OR NULL *SKIP LEADER OF (NULL...CR...LF) BEFORE EACH RECORD.	UNIMPLEMENTED	SKIP ONE STANDARD BINARY (OR ASCII) RECORD. TERMINATOR = COUNT IN BYTES 2, 3 OF RECORD. *SKIP LEADER OF (NUL...CR...LF) BEFORE EACH REC.	UNIMPLEMENTED
6 AVFILE	SKIP STD REC'S (AS ABOVE) TILL FILE MARK RECORD ($$) IS FOUND	UNIMPLEMENTED	SKIP STD. RECORDS (AS ABOVE) TILL FILE MARK RECORD ($$) IS FOUND	UNIMPLEMENTED
7 WEOF	null+	<------SAME	<---------SAME	<-------SAME
8 HOME	null+#	<------SAME	<---------SAME	<-------SAME

* This function defeated if bit 1 of UFT option word set.
\+ This operation sets EOF and BOM event bits in UFT status word.
\# Enables clear characters in hardware buffer on next READ or ADVANCE.

Table 3-1. Paper Tape Reader Operations and Modes

PROGRAMMER'S ERROR BIT IN UFT WORD UFTSTA	OPERATOR'S REASON CODE	REASON FOR ERROR IF SET
1	OVF OVERFLOW/ UNDERFLOW	-NONE-
2	PAR PARITY OR DATA CHECK	-NONE-
3	INO INOPERABLE	-DEVICE FAILED TO INTERRUPT AFTER OPERATION WAS INITIATED, CAUSING THE WATCHDOG TIMER TO EXPIRE.
4	MPE MEMORY PARITY OR IOP ERROR	-INVALID ADDRESS PARAMETERS WERE PASSED. (I.E., BUFFER OUTSIDE OF ADDRESSING RANGE. LOCAL RECOVERY ONLY. SYSTEM RECOVERY CAUSES THE TASK TO BE ABORTED.)
5	LOK WRITE LOCK	-NONE-
6	CIG CHARACTER IN GAP	-NONE-
7	OTH OTHER	-NONE-
8	SBV STANDARD BINARY VIOLATION OR DMP ERROR	STANDARD BINARY RECORD READ WAS LARGER THAN PROGRAMMER'S BUFFER. (NOT REPORTED TO OPERATOR).
PROGRAMMER'S EVENT BIT IN UFT WORD UFTSTA	REASON CODE	REASON FOR EVENT IF SET
9	EOM END-OF MEDIUM	-NONE-
10	EOF END-OF FILE	-A "$$" WAS ENCOUNTERED -A NULL OPERATION WAS REQUESTED -A REWIND OR WEOF WAS REQUESTED
11	BOM BEGINNING- OF-MEDIUM	-A REWIND WAS REQUESTED -A NULL OPERATION WAS REQUESTED

Table 3-2. Paper Tape Reader Errors and Events

CHAPTER 4
MAX IV PAPER TAPE PUNCH HANDLER (TPn,PPn)

The slow-speed paper tape punch (TPn) shares a channel with the teletypewriter. The high-speed punch (PPn) uses its own controller and address (group/unit) but operates in a similar manner. Both punches use the standard teletypewriter handler.

4.1 I/O OPERATIONS AND MODES

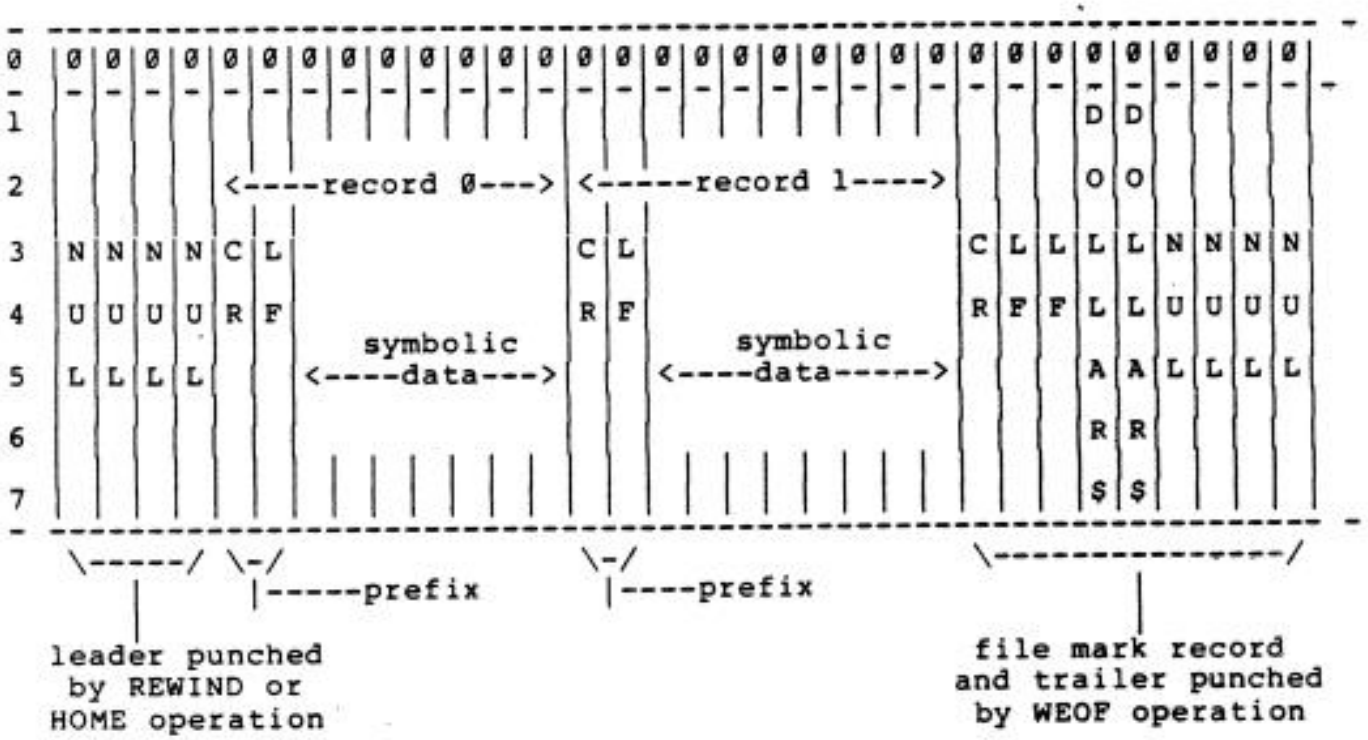
4.1.1 ASCII WRITE TO PAPER TAPE PUNCH (TPn,PPn)

The teletypewriter punch is physically indistinguishable from the teletype printer except for a "punch on" switch. However, it is logically distinguished because it is used as a secondary storage device instead of as a printing device. Likewise, its unit record size is indefinite (but not exceeding 32,768 bytes) while the printer is often limited to one carriage line (72 bytes). Carriage control bytes are not interpreted, since it is not a printing device, and are simply passed as data. A 7-bit USASCII code is transmitted to the punch for every 8-bit byte in the programmer's buffer (most significant bit always reset).

In the Standard ASCII mode, all WRITE operations have a carriage return (CR) and line feed (LF) byte prefixed to the data record being transmitted. This prefix serves as a unit record delineator when read by the paper tape reader (PRn) handler in the Standard ASCII or Binary modes and is not transmitted as data. An advantage of this prefix is that these secondary storage records of symbolic data can be listed offline in a consistent manner on a teletypewriter.

No prefix is transmitted in the Non-Standard ASCII mode, allowing records to be generated that interface with computers other than MODCOMP. Unit records transmitted in this mode must either be a fixed-block size or be terminated by a particular byte known to the programmer who must read them. A provision for termination of the WRITE, when a particular byte is found in the programmer's data buffer, is included in the mode. This byte, normally a NUL byte, can be specified to be any byte placed in the option word of the UFT. This byte is not transmitted. Refer to Figure 4-1.

Typical TPn,PPn Output of
Standard ASCII Records



Typical PTn,PPn Output of
Non-Standard ASCII Records

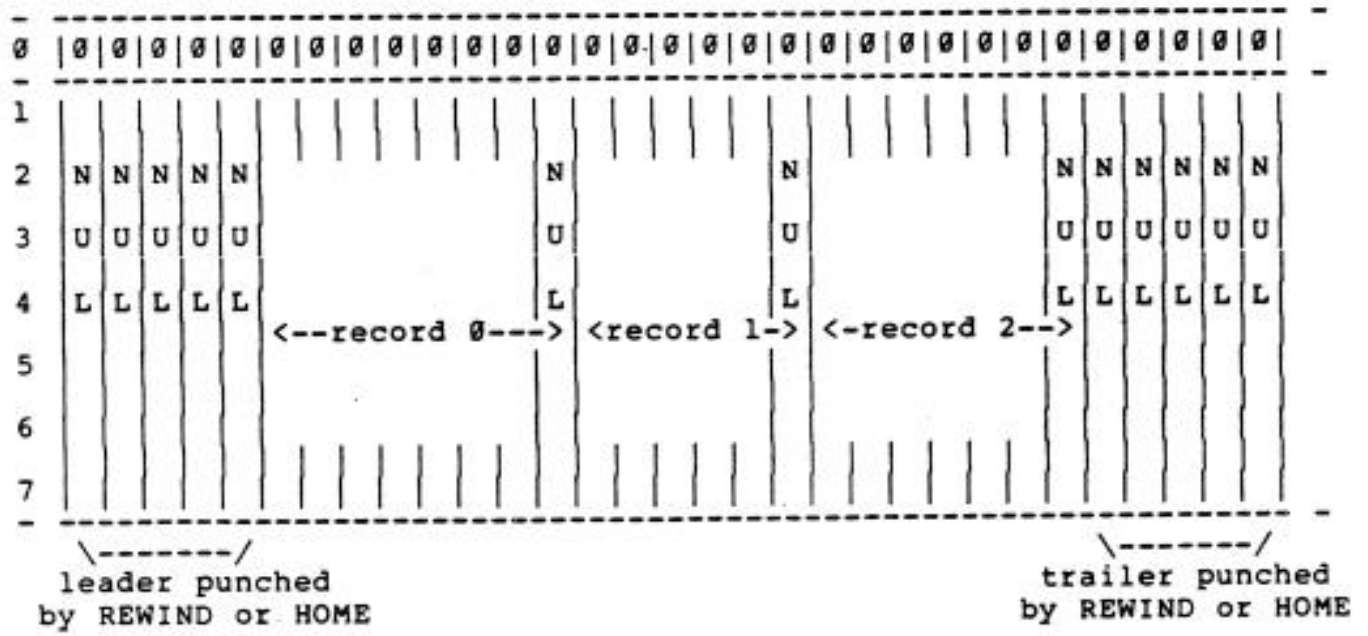


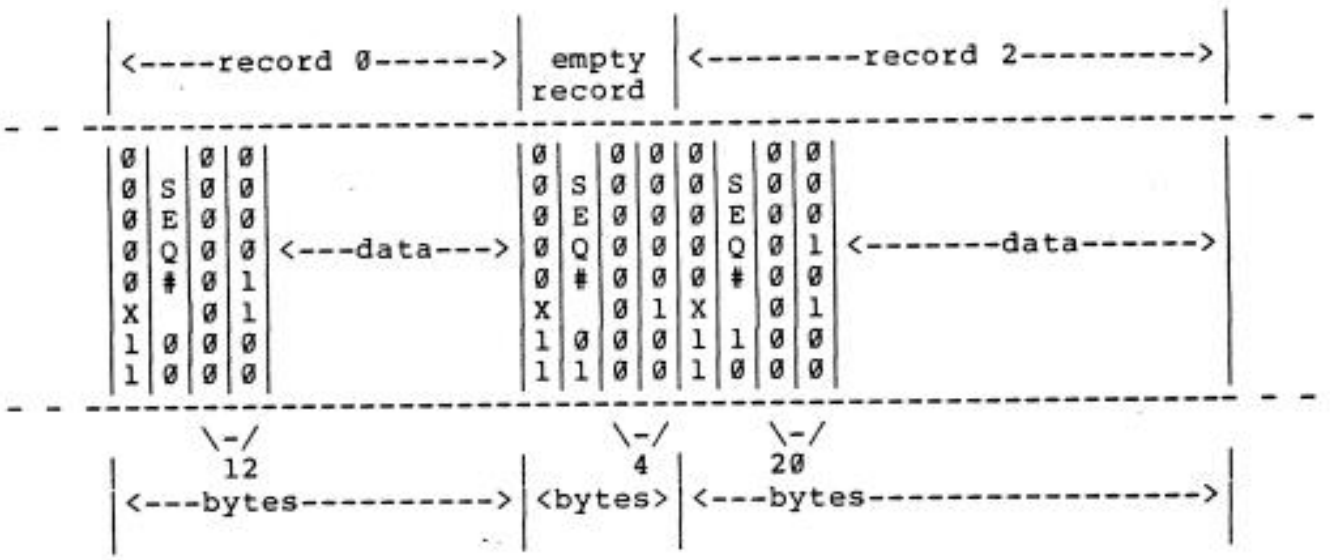
Figure 4-1. Punched Paper Tape Records (ASCII Formats)

4.1.2 BINARY WRITE TO PAPER TAPE PUNCH (TPn,PPn)

This mode transmits 8-bit binary data bytes to the punch. The Standard Binary mode requires that the data in the programmer's data buffer conform to the Standard Binary record whose size in bytes must be specified in bytes 2 and 3 of the data record. Byte 0 in the programmer's data buffer must be a binary record indicator byte (value of '00000011 or '00000111). Any attempt to transmit a record that does not conform to these conventions aborts the task with the reason code SBV (Standard Binary Violation).

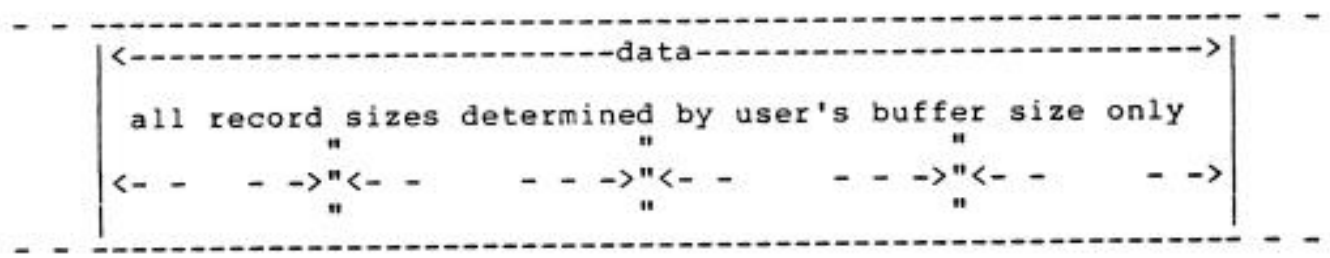
The Non-Standard Binary mode is similar to the Non-Standard ASCII mode except that an 8-bit data path is used, instead of the 7-bit ASCII data path, and only fixed-block length records can be transmitted. Refer to Figure 4-2.

Standard Binary Records - variable length



NOTE: X may be = 0 or = 1 in byte 0.
Leader may separate each record.

Non-Standard Binary Records - fixed length



all record sizes determined by user's buffer size only

Figure 4-2. Punched Paper Tape Records (Binary Formats)

4.1.3 FILE/RECORD OPERATIONS

4.1.3.1 REWIND and HOME to Paper Tape Punch (TPn,PPn)

The REWIND and HOME operations cause 256 bytes of leader to be transmitted to the punch, regardless of mode. Leader consists of NUL bytes ('00000000). The REWIND operation additionally resets the device position index to zero for the file and sets the EOF and BOM bits in the programmer's UFT.

4.1.3.2 WEOF to Paper Tape Punch (TPn,PPn)

This operation transmits a unit record sequence consisting of:

1. Carriage return (CR)
2. Two line feeds (LF)
3. Two dollar signs ($$)

This sequence is then followed by 256 bytes of trailer consisting of NUL ('00000000) bytes. The operation is appropriate for terminating a paper tape that must be torn from the punch.

The EOF bit is set in the programmer's UFT. The carriage return and line feed bytes act as leader if the file mark is read in any Standard ASCII or Binary mode. Only the $$ characters are significant data characters. A programmer may also produce valid file-mark records directly with the WRITE command if the buffer contains the $$ characters in word 0. Additional text may be transmitted after the $$ bytes and will be transmitted on a READ operation. Such an annotated file mark may be useful in some applications, but it is a device-dependent function. This technique does not apply to magnetic tape or disc, where file marks do not contain data.

The ASCII formats for punched paper tape records are illustrated in Table 4-1.

4.1.3.3 Null Operations to Paper Tape Punch (TPn,PPn)

The READ, BACKSPACE, and ADVANCE operations are not implemented and perform as null operations if requested. The EOF and BOM event bits are set in the programmer's UFT.

4.1.3.4 Summary Tables

Paper Tape Punch Handler summary tables appear at the end of this chapter. Table 4-1 contains information about the handler's response to I/O operations in the four modes of data transfer. Table 4-2 lists the error and event bits in the UFT status word (UFTSTA). Conditions causing the error or event are also listed.

4.2 HOLDUP CONDITIONS REPORTED TO OPERATOR

A message reporting a holdup condition is output to the operator in the following format:

```
!PPn?/tttttt/oooooo
!TPn?/tttttt/oooooo
```

tttttt - is the calling task's name

oooooo - is the current task overlay name

This message reports the following conditions:

- Device does not exist
- Device under exclusive use by another task
- Device already offline

4.3 SYSTEM GENERATION

The following statements are required in a system generation source file if the device is to be included in a system. Detailed descriptions of these statements and the rules of order in which they must be entered are included in the MAX IV SYSGEN, System Guide Manual.

```
CONTROLLER      PPx,#9      for high-speed punch (PP) device
    or
CONTROLLER      TTx,#A      for slow punch (TP) device on
                              teletypewriter
      .
      .
      .
DEVICE          PPn,PPx
        or
DEVICE          TPn,TTx
```

NOTE: The controller for the slow reader (TT) is the same controller defined for the TYn device. This controller statement should only be specified once.

PAPER TAPE PUNCHES (TPn OR PPn) RECORD SIZE ≤ 32768 BYTES

	<----------ASCII MODES---------->		<----------BINARY MODES---------->	
	STANDARD	NON-STANDARD	STANDARD	NON-STANDARD
0 READ	null+	<-------SAME	<-------SAME	<-------SAME
1 WRITE	OUTPUT ONE STANDARD ASCII RECORD WITH (CR, LF) LEADER (STOP TRANS-MISSION IF (NUL) BYTE IN BUFFER)	OUTPUT ONE USER-DEFINED RECORD; NO LEADER PREFIX (STOP TRANS-MISSION IF USER-DEFINED BYTE IN BUFFER)	OUTPUT ONE STD. BINARY RECORD (MUST CONFORM TO STD.BINARY FORMAT) (STOP TRANSMISSION WHEN BYTE COUNT IN BUFFER BYTE 2, 3 EXCEEDED)	OUTPUT ONE BLOCK (RECORD = SIZE OF USER'S BUFFER)
2 REWIND	PUNCH 256 (NUL) BYTES OF LEADER	<-------SAME	<-------SAME	<-------SAME
3 BKFILE	null+	<-------SAME	<-------SAME	<-------SAME
4 BKRECORD	null+	<-------SAME	<-------SAME	<-------SAME
5 AVRECORD	null+	<-------SAME	<-------SAME	<-------SAME
6 AVFILE	null+	<-------SAME	<-------SAME	<-------SAME
7 WEOF	OUTPUT FILE MARK ($$) RECORD WITH (CR,2LF) LEADER AND 256 (NUL) BYTES OF TRAILER	<-------SAME	<-------SAME	<-------SAME
8 HOME	PUNCH 256 NUL BYTES OF LEADER	<-------SAME	<-------SAME	<-------SAME

+ This operation sets the EOF and BOM event bits in the UFT status word.

Table 4-1. Paper Tape Punch Operations and Modes

PROGRAMMER'S ERROR BIT IN UFT WORD UFTSTA	OPERATOR'S REASON CODE	REASON FOR ERROR IF SET
1	OVF OVERFLOW/ UNDERFLOW	-NONE-
2	PAR PARITY OR DATA CHECK	-NONE-
3	INO INOPERABLE	-THE DEVICE FAILED AFTER AN OPERATION FAILED, CAUSING THE WATCHDOG TIMER TO EXPIRE.
4	MPE MEMORY PARITY OR IOP ERROR	-INVALID ARGUMENTS WERE SPECIFIED IN A WRITE CALL. (I.E., BUFFER ADDRESS OUTSIDE OF ADDRESSING RANGE.) SYSTEM RECOVERY CAUSES THE TASK TO ABORT.
5	LOK WRITE LOCK	-NONE-
6	CIG CHARACTER IN GAP	-NONE-
7	OTH OTHER	-NONE-
8	SBV STANDARD BINARY VIOLATION	-A STANDARD BINARY RECORD IS BEING WRITTEN WHOSE BYTE COUNT IS LARGER THAN THE OUTPUT BUFFER, OR THE RECORD DOES NOT BEGIN WITH #03 OR #07.
PROGRAMMER'S EVENT BIT IN UFT WORD UFTSTA	REASON CODE	REASON FOR EVENT IF SET
9	EOM END-OF- MEDIUM	-PUNCH TAPE LOW (HIGH SPEED ONLY)
10	EOF END-OF- FILE	-A WEOF WAS REQUESTED -A REWIND WAS REQUESTED -A NULL OPERATION WAS REQUESTED
11	BOM BEGINNING- OF-MEDIUM	-A REWIND WAS REQUESTED -A NULL OPERATION WAS REQUESTED

Table 4-2. Paper Tape Punch Errors and Events

CHAPTER 5
MAX IV CARD READER HANDLER (CRn,DRn)

This chapter describes two card reader handlers. One handler transmits data into memory using programmed data interrupts and is "universal" in that it will operate on all Card Reader Controllers. The other handler supports only the newer models and transmits data into memory using a Direct Memory Processor (DMP) channel. To MAX IV, the data interrupt handler and its controller are known by the name 'CR' and the DMP handler and its controller are known by the name 'DR'. When the DMP controller is available, the use of the DMP handler is recommended because it performs work more efficiently and is immune to data overflow errors that can occur with the data interrupt handler in certain interrupt environments.

The standard group/unit number of the I/O bus channel is 5 for either controller. The standard DMP channel number is also 5 when it is used. Refer to the section on SYSTEM GENERATION for methods of choosing the desired handler. Each handler can serve more than one controller when required.

Both handlers are capable of supporting the reading of cards in all standard (device-independent) access methods as well as all non-standard (device-dependent) access methods. There is a provision for reading symbolic data in ASCII code, using either Ø29 or Ø26 card-punch character sets and binary data in standard MODCOMP Binary or 12-bit row and column format.

5.1 A BRIEF DESCRIPTION OF HANDLER OPERATION

This section describes some of the internal operations that take place in the handler after an application programmer has requested the reading of a record from some logical file that happens to be assigned to the Card Reader Logical Device.

The request to read is routed through a BASIC I/O SYSTEM (BIOS) REX Service called directly by the user. This request is queued until the card reader controller is capable of performing the operation. When ready, the controller is commanded to read a card. When using the data-interrupt controller, as each card column passes the read head, a data-interrupt is requested. The handler element that processes data-interrupts takes the data representing the 12-bit card column (or an 8-bit compressed form called "transition code") and either stores it or translates it to a suitable symbolic code by software table lookup before storing it.

The MAX IV Input Spooling Symbiont is used to interpose high-speed buffering between a task trying to read from logical device CR, and the actual card reader. The reader is referred to the chapter on the input spooler in this manual for detailed information.

When using the DMP handler/controller the 12-bit card codes for each character are stored in an intermediate memory called the data RAM in the controller. While this memory is being filled with the entire contents of the card, the DMP channel can also begin to transmit directly into the memory area specified in the caller's program. Before each character transmission, however, the controller may choose to translate the 12-bit code into one of several other requested codes by various hardware translation methods.

After all the data has been transmitted, the handler makes certain determinations about the number of characters transmitted (access method dependent), completes the operation, and signals the requesting program of its completed operation.

Code translation is similar in both handlers, although different hardware and software techniques are used in each case. Figure 5-1 summarizes the data transformations that take place for each character.

The 12-bit raw character from the card is either stored (one per 16-bit word - right justified) as is, or translated to "transition" code. This transition code is achieved by passing rows 12, 11, 0, 8, and 9 directly, and then "compressing" rows 1-7 into a 3-bit binary field. This is possible only because standard symbolic Hollerith codes for cards never have more than one hole punched in rows 1-7 for any valid character. Thus, the unitary 7-bit field can be easily converted into a binary 3-bit field whose value indicates "which", if any, bit is punched. This "Transition" code is used as a "binary" data format by MODCOMP software instead of the 12-bit code.

The 8-bit transition code is either stored, (two per 16-bit word) as is, or translated further into an internal "symbolic" character code such as ASCII. In the data interrupt handler, this is done by software using the "transition code" as an index to a memory-resident table of character codes. The code thus fetched from the table is used instead. In the DMP controller, this translation is performed by a hardware translation memory called the translation RAM. The DMP handler automatically loads this special memory in the controller with a suitable translation table each time a new table is needed or requested explicitly (DDO option bit in caller's UFT or task's option bit U6). Most ASCII translated codes are 7-bits in length. They are stored two per 16-bit word, with each 7-bit code right-justified in each 8-bit byte (bit 0 of each byte is '0' for all valid characters).

Additional considerations are discussed for each access method (mode) in the following sections.

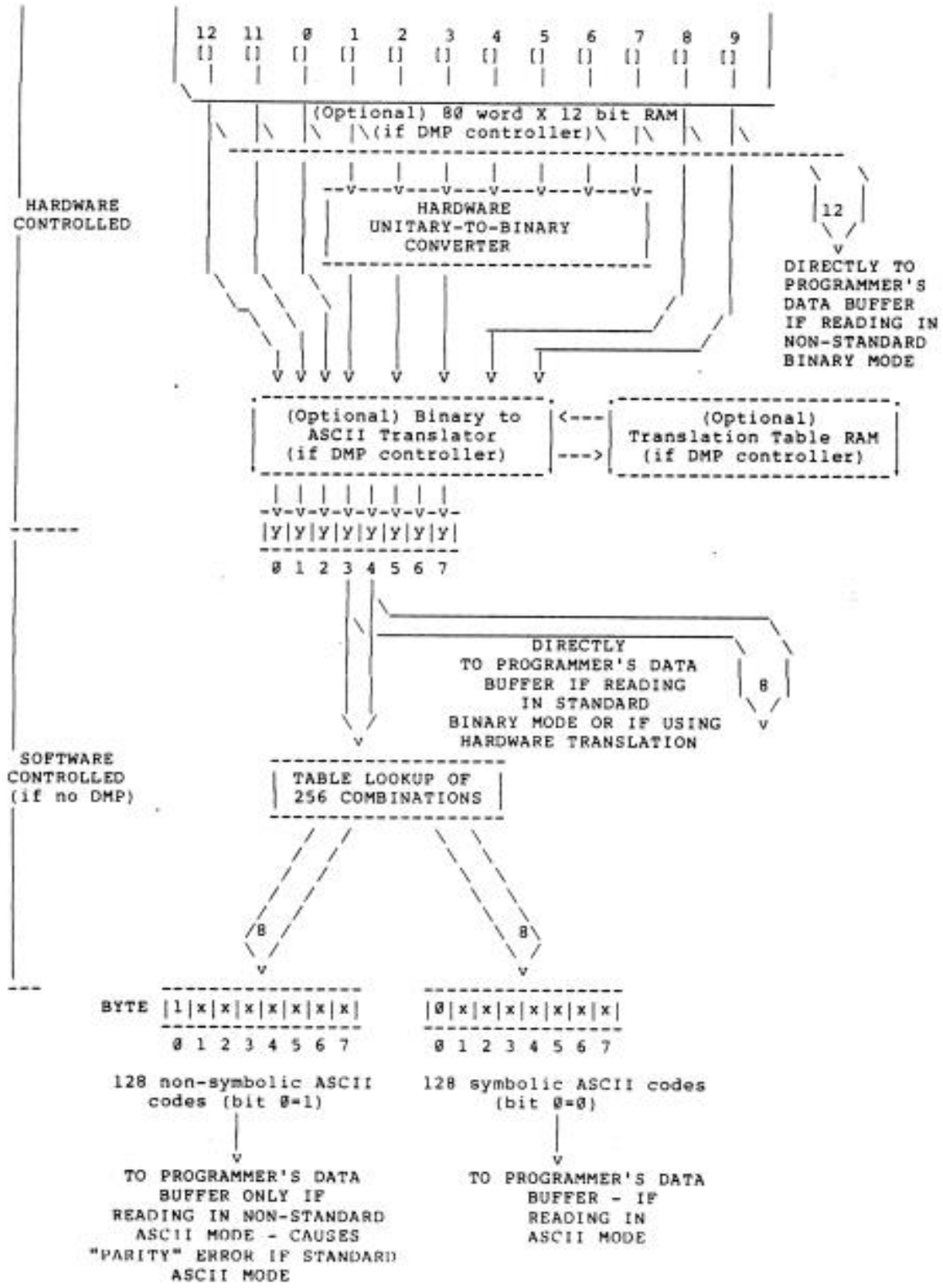


Figure 5-1. Card Reader Data Paths from Card to Memory

5.2 I/O OPERATIONS AND MODES

5.2.1 STANDARD ASCII READ FROM CARD READER (CRn)

The card reader transmits one card as a unit record (80 bytes) for each queued READ operation. Data buffers exceeding 40 words will be truncated to 40 words by the handler.

In this mode, transition codes are read from the card controller; they are translated into a valid USASCII code by indexed table lookup, (non-DMP) or will be translated by the hardware RAM and then transmitted (DMP) into memory. In non-DMP mode, translation occurs by software, at the data interrupt level, which receives each card column.

The handler has the provision for two translation tables, selection being controlled at run-time by the status of a task option bit (U6) or bit 1 (DDO) of the UFT option word (Word 2). If a task reading a card has option U6 reset and DDO reset, the normal translation table is used (ANSI Hollerith card code to ASCII). The ANSI Hollerith card code is similar to the IBM 029 card code, except for four characters:

ANSI Card Character	029 Character	Internal ASCII Code
exclamation point	vertical bar	#21
left bracket	cent sign	#5B
right bracket	exclamation point	#5D
caret (up arrow)	underline	#5E

5.2.1.1 Alternate Translation Table

If option U6 in the task option word is set or DDO in the caller UFT is set, an alternate translation table is used. This alternate table is externally defined to the handler and defaults to the normal table if no special action is taken at system generation. However, if an alternate table is supplied as a system element at SYSGEN, then it becomes the alternate table.

A table for converting Ø26-keypunch codes into a partial ASCII set is available at system generation time (see CARDCODE SYSGEN macro in the MAX IV SYSGEN, System Guide Manual). The format of this table is very simple; however, the user may wish to build one for other code conversions:

```
((CRxxxx
            TTL       ALTERNATE CARD CONVERSION TABLE
            INT       CR$CVT
CR$CVT      DFC       #aabb \
            DFC       #ccdd  \  128 words or 256 bytes ordered
             .          .    \ in ascending value of raw 8-bit
             .          .    / column code
             .          .   /
            DFC       #ppqq /
            END
```

NOTE: (xxxx) is the 1- to 4-character name of the insert specified CARDCODE macro. Symbol (CR$CVT) must be specified as the table's first word.

Place into each byte position (aa, bb, ..., qq) the internal code that is desired when the appropriate "transition" code (#ØØ, #Ø1, ..., #FF) is received. All these bytes should have the left-most bit reset. The remaining positions that do NOT correspond to valid USASCII codes should contain unique codes with the left-most bit set. Such invalid column codes force a card READ error (PARITY) and termination of the READ (data interrupt handler only). The "invalid" translated codes are passed into memory without error if the Non-Standard ASCII mode is used or if the DMP handler is used.

5.2.1.2 Editing Features

The Standard ASCII READ mode has several editing features, which support standard software. Trailing blanks on the card are "suppressed", and the byte count returned to the caller's UFT adjusted accordingly. In addition, the "end-of-useful data" is marked by placing a zero word (2 NUL bytes) in the user's data buffer to mark the first whole word where trailing blanks originally started. The marker gives the basically fixed-length card record a variable-length nature. When free-form symbolic data is punched on a card, the NUL marker provides syntax analysis routines with a convenient terminator so that they do not have to limit searches to unit record size. When the possibility exists that all 8Ø card columns (4Ø words) will contain useful data, the programmer should define the buffer to be one word longer than the card image:

```
BUF   RES 41,Ø
```

But the call should specify the true card size:

```
LDI,8      READ
LDI,14     BUF
LDI,15     8Ø
REX,MAXIV
```

This ensures that the NUL marker (Word 41 = Ø) at the end of the buffer will always be present to mark the end-of-useful information. This is necessary because the handler never stores the NUL marker if it exceeds the buffer bounds specified by the call.

If a Standard Binary card format is detected (column 1 = ETX (#Ø3) or BEL (#Ø7) codes) while reading in the Standard ASCII mode, the entire card is read in the Binary mode and an indicator bit (Bit 8 in the UFT status word) is set without an error condition being reported. This allows intermixed Binary and ASCII cards to be read by special programs without knowing when specific cards are to appear.

Upon completion of a Standard ASCII READ, the number of bytes of useful data (8Ø bytes minus the number of trailing blanks (filling whole words)) is stored into the UFT for use by the program.

5.2.2 NON-STANDARD ASCII READ FROM CARD READER (CRn)

The Non-Standard ASCII mode does not suppress blanks, nor does it mark the end-of-useful information with a zero word. However, it has a feature that allows the end-of-useful information from the card to be marked whenever a particular ASCII byte is received that matches the byte stored in the programmer's UFT option word. This byte is normally a NUL byte but can be any 7-bit ASCII character. The terminating byte is stored in the data buffer. If the byte is not detected on the card's data stream, the transmission terminates at the end-of-card or when the specified number of bytes has been transmitted.

The UFT will contain the actual number of bytes transferred between the start of the card and the terminating character. Either the normal or ALTERNATE ASCII conversion table may be specified as with Standard ASCII. Parity Errors (ASCII byte with the most significant bit set) are not logged when using this mode.

5.2.3 STANDARD BINARY READ FROM CARD READER (CRn)

The Binary mode causes each 8-bit "transition" code derived by hardware conversion from the 12-bit column, to be stored directly in the programmer's buffer. The Standard Binary mode requires that the first column on the card contain the Standard Binary record indicator (either #Ø3 or #Ø7) and that columns 3 and 4 contain a byte count that defines the record size (<=8Ø). If this information is not found, an internal switch to Standard ASCII mode occurs and the card is read in this mode. A status bit, not identified as an error, is set in the programmer's UFT status word (Bit 8). This mode is used to read any standard object code produced on a card punch with the Assemblers, the Library Update program, the MAX III/IV Link Editor, TOC, et cetera.

5.2.4 NON-STANDARD BINARY READ FROM CARD READER (CRn)

This Binary mode passes all 12-bit "raw" column codes directly to the programmer's data buffer in a format shown in Figure 5-2. This method requires an 80-word (160-byte) buffer for each card instead of the 40-word (80-byte) buffer required for other data modes. For the non-DMP Handler, the maximum length of the DEVICE (Word 0 of the LDT) must be specified as 160 bytes at system generation time. (See the DEVICE statement described later in this document.) This is not necessary for the DMP Handler. The DMP handler's LDT must be specified as 40 words (80 bytes).

CARD ROW / BIT	0	1	2	3	12 4	11 5	0 6	1 7	2 8	3 9	4 10	5 11	6 12	7 13	8 14	9 15
WORD 0	0	0	0	0	COLUMN 1											
1	0	0	0	0	COLUMN 2											
2	0	0	0	0	COLUMN 3											
3	0	0	0	0	COLUMN 4											
.	.		.		.						.					
.	.		.		.						.					
.	.		.		.						.					
79	0	0	0	0	COLUMN 80											

Figure 5-2. Packing Raw Columns into Memory

5.2.5 FILE/RECORD OPERATIONS

5.2.5.1 File Mark Detection

On READ or ADVANCE operations, regardless of the STANDARD data mode used, any card encountered whose first two columns contain the translated bytes "$$" is considered as a file-mark record and sets the EOF event bit in the programmer's UFT. If such a card has additional data, it is passed to the programmer's buffer, and translated to ASCII. No file mark detection is performed in the Non-Standard modes.

5.2.5.2 ADVANCE RECORD/FILE From Card Reader

These operations cause the reading of cards into a dummy buffer. Only enough of the card is read, however, to determine if it is a file-mark card. The EOF event bit is set in the UFT when a card, starting with "$$", is encountered. The ADVANCE RECORD operation causes only one card to be read. The ADVANCE FILE operation causes many cards to be read until a file-mark card is found. These operations should only be performed in the Standard ASCII or Standard Binary mode.

5.2.5.3 Null Operations from Card Reader

The WRITE, REWIND, HOME, WEOF, and BACKSPACE operations are not implemented and perform as null operations if requested. All of these operations set the EOF and BOM event bits in the programmer's UFT, if attempted. The REWIND operation resets the File Position Index in the UFT to zero.

5.2.5.4 Summary Tables

Card Reader Handler summary tables appear at the end of this chapter. Table 5-1 contains information about the handler's response to I/O operations in the four modes of data transfer. Table 5-2 lists the error and event bits in the UFT status word (UFTSTA). Conditions causing the error or event are also listed.

5.3 ERROR RECOVERY AND HOLDUP CONDITIONS

Both device handlers report errors directly to either the programmer or the system operator (at the programmer's option) whenever any error condition is detected. Refer to Table 5-2, Card Reader Errors and Events.

A message reporting a holdup condition is output to the operator in the following format:

```
!CRn?/tttttt/oooooo

      tttttt - is the calling task's name

      oooooo - is the current task overlay name
```

This message reports the following conditions:

- Card reader in hold state (not ready)
- Device under exclusive use by another task
- Device already offline

5.4 SYSTEM GENERATION

One CONTROLLER statement and one DEVICE statement are required in a system generation source file if this device is to be included in a system. Detailed descriptions of these statements and the rules of order in which they must be entered are included in the MAX IV SYSGEN, System Guide Manual.

```
CONTROLLER  CRn,5     use this statement if non-DMP handler
                      desired,
CONTROLLER  DRn,5,5   or use this statement if DMP handler
                      desired.
     .
     .
     .
DEVICE      CR,CRn    use this statement if non-DMP handler, or
DEVICE      CR,DRn    use this statement if DMP handler
```

Because of the nature of the HOLD switch on this device, its controller should not be timed by a watchdog timer. If additional card readers are added, the controller names and device names must be given a unique third character (for example, DR1 or CR1). The default record size of the device is normally 80 bytes, except when the column binary mode of read (Non-Standard Binary) is to be used with the non-DMP handler.

If an alternate 026 card code conversion table is desired, the following system generation statement is required somewhere in the I/O Block assembly:

```
CARDCODE   026
```

If the user has defined a special form of the alternate conversion table, then the following statement is required:

```
CARDCODE   xxxx
```

xxxx - is the last four characters of an insert name ((CRxxxx on which the table is defined in the macro insert file MC.

CARD READERS (CRn/DRn)

RECORD SIZE ≤ 160 BYTES FOR NON-STD. ASCII,
OR RECORD SIZE ≤ 80 BYTES FOR OTHER MODES

	<----------ASCII MODES---------->		<---------BINARY MODES-------->	
	STANDARD	NON-STANDARD	STANDARD	NON-STANDARD
0 READ	INPUT ONE STANDARD ASCII (OR BINARY) CARD WITH 029-(OR 026 OPTIONAL) TO-ASCII CONVERSION IF ASCII (SUPPRESS TRAILING BLANKS).	<-------SAME BUT DOES NOT SUPPRESS TRAILING BLANKS.	INPUT ONE STD. BINARY (OR ASCII) CARD W/ 12 BIT - 8 BIT TRANSITIONAL CODE CONVERSION (PACK 2 COLUMNS PER WORD)	INPUT ONE CARD INTO 80 WORD BUFFER STORING RAW 12-BIT COLUMN CODES PACKED 1-PER WORD RIGHT-JUSTIFIED
1 WRITE	null+	<-------SAME	<-------SAME	<-------SAME
2 REWIND	null+ Resets FPI in UFT to zero	<-------SAME	<-------SAME	<-------SAME
3 BKFILE	null+	<-------SAME	<-------SAME	<-------SAME
4 BKRECORD	null+	<-------SAME	<-------SAME	<-------SAME
5 AVRECORD	SKIP ONE CARD. CHECK FOR ($$) FILE MARK RECORD.	SKIP ONE CARD (NO FILE MARK TEST).	SKIP ONE CARD. CHECK FOR ($$) FILE MARK RECORD.	SKIP ONE CARD (NO FILE MARK TEST).
6 AVFILE	SKIP CARDS TILL ($$) FILE MARK RECORD FOUND	UNIMPLEMENTED	SKIP CARDS TILL ($$) FILE MARK RECORD FOUND	UNIMPLEMENTED
7 WEOF	null+	<-------SAME	<-------SAME	<-------SAME
8 HOME	null+	<-------SAME	<-------SAME	<-------SAME

+ This operation sets the EOF and BOM event bits in the UFT status word.

Table 5-1. Card Reader Operations and Modes

PROGRAMMER'S ERROR BIT IN UFT WORD UFTSTA	OPERATOR'S REASON CODE	REASON FOR ERRORS IF SET
0	ER	-SOME ERROR HAS OCCURRED AS PER BITS 1-8.
1	OVF OVERFLOW/ UNDERFLOW	-OVERFLOW- TWO COLUMNS PAST READ HEAD BEFORE DATA INTERRUPT COULD BE SERVICED BY HANDLER. CARD IS IN STACKER. OPERATOR SHOULD REPLACE IN HOPPER AND TELL SYSTEM TO RETRY. THE DMP HANDLER WILL NEVER DETECT THIS ERROR.
2	PAR PARITY OR DATA CHECK	-LIGHT/DARK CHECK FAILED. -CARD MOTION ERROR DETECTED. -ILLEGAL ASCII CONVERSION (TABLE LOOKUP) IN STANDARD ASCII MODE ONLY. BAD CARD IS IN STACKER. OPERATOR SHOULD REPLACE IN HOPPER AND TELL SYSTEM TO RETRY. REPUNCH OF CARD MAY BE NECESSARY. THE DMP HANDLER WILL NEVER DETECT THIS ERROR.
3	INO INOPERABLE	-DEVICE DOES NOT EXIST OR, -DEVICE INOPERABLE
4	MPE MEMORY PARITY	-MPE DETECTED DURING A DMP DATA TRANSFER. FOR DMP HANDLER, THIS CAN MEAN THAT THE PROGRAMMER HAS SPECIFIED BUFFER AREAS IN NON-EXISTENT OR READ-ONLY MEMORY AREAS OF ITS ADDRESSING SPACE.
5	LOK WRITE LOCK	-NONE-
6	CIG CHARACTER IN GAP	-MULTIPUNCH DETECTED IN ROWS 1-7 (DMP HANDLER ONLY).
7	OTH OTHER	-NONE-
8	HOL HOLD	-NONE-

PROGRAMMER'S EVENT BIT IN UFT	REASON CODE	REASON FOR EVENT IF SET
8 (WITH ERROR BIT = 0).	SBV STANDARD BINARY VIOLATION	-A STANDARD BINARY CARD WAS READ WHEN READING STANDARD ASCII MODE. -A STANDARD ASCII CARD WAS READ WHEN READING IN STANDARD BINARY MODE. -SOME OTHER LOGICAL PROGRAMMER ERROR.
9	EOM END-OF MEDIUM	-HOPPER EMPTY/STACKER FULL
10	EOF END-OF- FILE	-SET AFTER REWIND OR NULL OPERATIONS -SET AFTER $$ DETECTED IN STANDARD MODES.
11	BOM BEGINNING- OF-MEDIUM	-SET AFTER REWIND OR NULL OPERATIONS.

Table 5-2. Card Reader Errors and Events

CHAPTER 6
MAX IV CARD PUNCH HANDLER (CPn)

The card punch handler consists of the standard three levels of execution. These are the TASK level and directly connected SERVICE and DATA interrupt levels utilizing unique hardware party-line interrupt vectors capable of controlling the transfer of data, status, and commands to/from a card punching device.

The hardware controller supports only one punch. However, additional punches may be added to additional hardware controllers if each is assigned a unique group/unit number. A single copy of the handler will serve all of them.

The standard card punch has the controller name CP and uses group/unit #06. No DMP (Direct Memory Processor) hardware is required since the data rate is sufficiently low to use data interrupts for each transmitted column. The standard logical device name is CP and must be assigned to controller CP at system generation. Additional punches must be assigned controller and logical device names of the form CPn, where n is a unique character distinguishing each controller/device pair.

6.1 I/O OPERATIONS AND MODES

6.1.1 ASCII WRITE TO CARD PUNCH (CPn)

The programmer's data buffer is transmitted to the punch as 7-bit ASCII characters packed two per word. The punch hardware converts these codes into an 8-bit transitional code by a hardware table lookup using a ROM (Read-Only Memory). The resulting compressed code is further translated into a 12-bit column that produces a standard Hollerith card punch code (029-keypunch code). The method is the reverse of the card READ operation, only a hardware conversion is used instead of a software method.

An 80-byte card is the standard unit record size for data transmission (40 words maximum). Any larger amount of data in the programmer's buffer is not transmitted. Any carriage control bytes in bytes 0 and 1 of the buffer are transmitted as data and are not interpreted as they would be in a printing device. Data bytes are transmitted until a null (NUL) byte marker is found in the programmer's buffer, or until 80 bytes are transmitted, or until the specified buffer size is exceeded. Physical records are always fixed length; but, transmission length can be determined by call arguments or by the contents of the data buffer, whichever is more convenient to the programmer. Cards punched in this data mode should not contain control bytes such as NUL, CR, and LF since they cause the complementary card READ mode to operate in unusual ways when such unit records are processed in a device-independent manner.

A Non-Standard ASCII WRITE mode also stops transmission when a NUL byte is encountered in the data buffer; however, the programmer may put any valid ASCII bytes in the option word of the UFT and this byte will terminate transmission instead of the NUL byte. If bit 8 of the UFT option word is set, the premature termination feature is defeated. The terminating byte is not transmitted to the card.

6.1.2 BINARY WRITE TO CARD PUNCH (CPn)

This mode transmits 8-bit binary data bytes, packed two per word, to the punch. Each byte is converted directly into a 12-bit column by the method shown in Figure 6-1. The Standard Binary mode imposes restrictions on the format of the first four data bytes in the programmer's data buffer, requiring that byte Ø be a Standard Binary record indicator (#Ø3 or #Ø7) and that bytes 2 and 3 define the record size in bytes (<=8Ø). The remaining bytes may be data. Any attempt to transmit a Standard Binary record not conforming to this convention aborts the task with the reason code SBV (Standard Binary Violation). This convention is not required in the Non-Standard Binary mode, which transmits all the data to the card unmodified.

6.1.3 FILE/RECORD OPERATIONS

6.1.3.1 WEOF to Card Punch (CPn)

This WRITE operation punches a card with the Hollerith characters "$$" punched in columns 1 and 2. Although a Standard ASCII (Hollerith Ø26 or Ø29) record, this card is also detected as a file mark when read in a Standard Binary or Standard ASCII mode by the card reader handler. Such a record can also be written with the conventional WRITE service when additional text is desired on the "file mark" record.

6.1.3.2 REWIND and HOME to Card Punch (CPn)

These operations do not transmit any data; however, they cause a status bit to be set in the card punch controller table that causes the next WRITE or WEOF operation to offset the punched card in the stacker. (Some card punches do not perform this offset function.) This allows many punch jobs to be processed without operator intervention until the stacker is full. The operator can then separate the many jobs by pulling them apart at each offset (initial) card. REWIND additionally sets the device position index of the file to zero and sets the EOF and BOM event bits in the UFT status word.

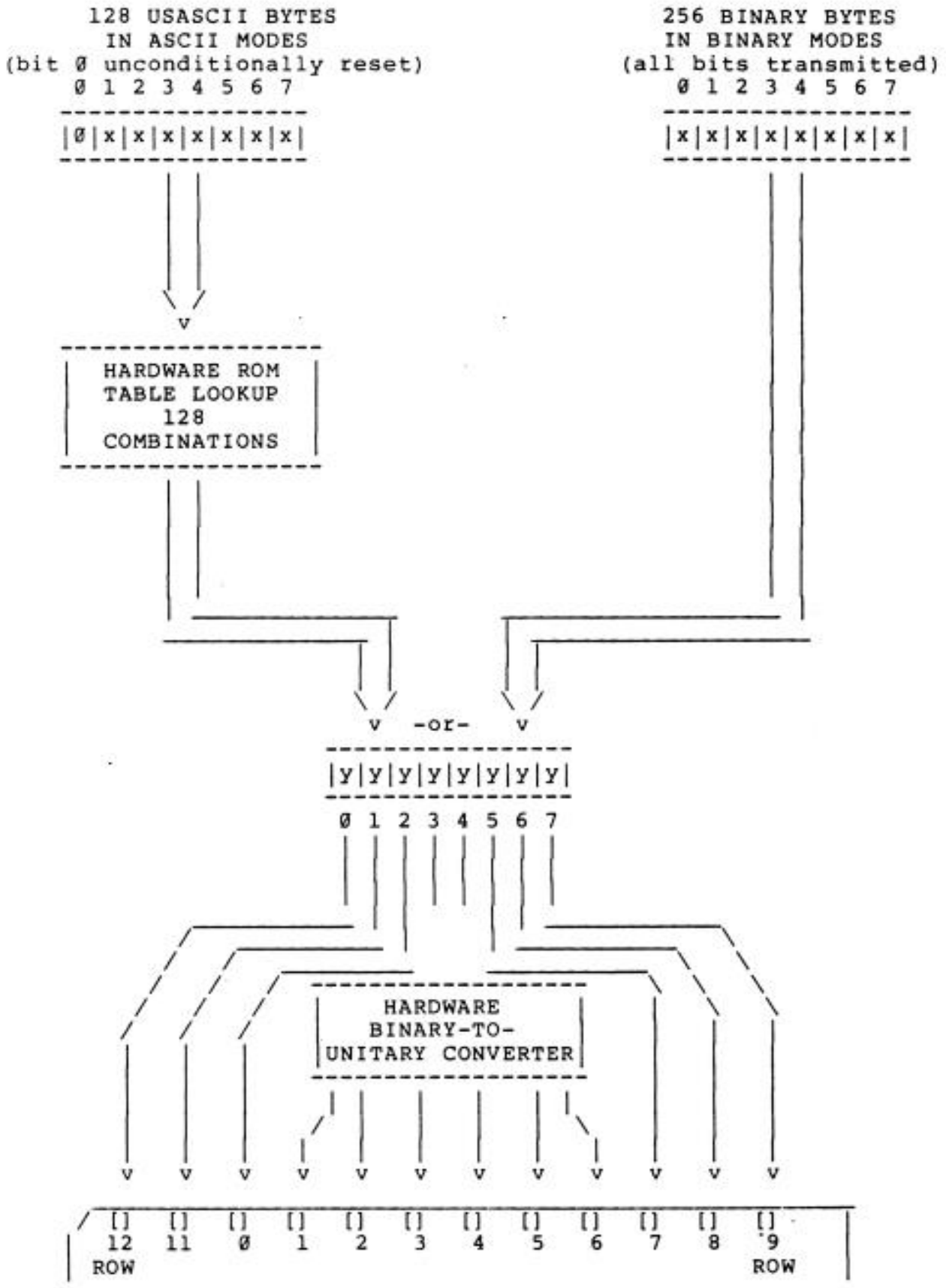


Figure 6-1. Card Punch Data Paths from Memory to Card

6.1.3.3 Null Operations to Card Punch (CPn)

The READ, BACKSPACE, and ADVANCE operations are not implemented and perform as null operations if requested. The EOF and BOM event bits are set in the programmer's UFT.

6.1.3.4 Summary Tables

Card Punch Handler summary tables appear at the end of this chapter. Table 6-1 contains information about the handler's response to I/O operations in the four modes of data transfer. Table 6-2 lists the error and event bits in the UFT status word (UFTSTA). Conditions causing the error or event are also listed.

6.2 SPECIAL UNIMPLEMENTED HARDWARE FEATURE

A hardware command that inhibits the conversion from 8-bit data bytes to 12-bit columns is accepted by the standard hardware controller. This feature opens up a direct 12-bit data path allowing a handler to transmit true multipunch column binary data. The present card punch handler does not implement this feature since this binary format is not used by MODCOMP Standard Software. Such a feature could be added to the handler by the user if the application requires it. The Non-Standard Binary mode is a suitable vehicle for this implementation. The modified handler could pass right-justified 12-bit fields directly from each word of the programmer's data buffer, or it could unpack the data in 12-bit fields. Refer to the MAX IV CARD READER HANDLER, a chapter in this manual.

6.3 ERROR RECOVERY AND HOLDUP CONDITIONS

This device handler, like all devices, reports all errors to the programmer, or to the system (operator) as the programmer options. No error conditions institute automatic retry procedures. All error conditions described below are reported immediately upon detection:

1. Device inoperable (for example, power off, device not ready).
2. Data rate overflow/underflow.
3. Parity (echo check data error).
4. Watchdog timer expiration (.12 seconds after last operation initiated).

In response to Error 1, the operator usually readies the device and continues. For Errors 2 and 3, the card has already been punched so the operator should remove it from the stacker before the device is allowed to continue.

A message reporting a holdup condition is output to the operator in the following format:

```
!CPn?/tttttt/oooooo
```

tttttt - is the calling task's name

oooooo - is the current task overlay name

This message reports the following conditions:

- Device does not exist
- Device under exclusive use by another task
- Device already offline

6.4 SYSTEM GENERATION

The following statements are required in a system generation source file if this device is to be included in a system. Detailed descriptions of these statements and the rules of order in which they must be entered are included in the MAX IV SYSGEN, System Guide Manual.

```
CONTROLLER          CPn,6
     .
     .
     .
DEVICE              CP,CPn
```

If additional card punches are added, the controller names and device names must be given a unique third character (for example CP2).

CARD PUNCHES (CPn) — RECORD SIZE ≤ 80 BYTES

	ASCII MODES		BINARY MODES	
	STANDARD	NON-STANDARD	STANDARD	NON-STANDARD
0 READ	null+	<-------SAME	<-------SAME	<-------SAME
1 WRITE	OUTPUT ONE CARD WITH ASCII-TO-029 HARDWARE CONVERSION (STOP TRANSMISSION IF NULL IN BUFFER)	OUTPUT ONE CARD WITH ASCII-TO-029 HARDWARE CONVERSION (STOP TRANSMISSION IF USER-DEFINED BYTE IN BUFFER)	OUTPUT ONE STD. BINARY CARD WITH 8 BIT TRANSITIONAL CODE CONVERSION (STOP TRANSMISSION WHEN BYTE COUNT IN BUFFER BYTES 2, 3 EXCEEDED)	OUTPUT ONE CARD TRANSITIONAL 8 BIT-TO-12 BIT CONVERSION (80 BYTE RECORDS ONLY)
2 REWIND	null+ (BUT WILL OFFSET THE NEXT CARD PUNCHED)	<-------SAME	<-------SAME	<-------SAME
3 BKFILE	null+	<-------SAME	<-------SAME	<-------SAME
4 BKRECORD	null+	<-------SAME	<-------SAME	<-------SAME
5 AVRECORD	null+	<-------SAME	<-------SAME	<-------SAME
6 AVFILE	null+	<-------SAME	<-------SAME	<-------SAME
7 WEOF	OUTPUT FILE MARK ($$) CARD -- REMAINDER OF CARD IS BLANK	<-------SAME	<-------SAME	<-------SAME
8 HOME	null+ (BUT WILL OFFSET THE NEXT CARD PUNCHED)	<-------SAME	<-------SAME	<-------SAME

+ This operation sets the EOF and BOM event bits in the UFT status word.

Table 6-1. Card Punch Operations and Modes

PROGRAMMER'S ERROR BIT IN UFT WORD UFTSTA	OPERATOR'S REASON CODE	REASON FOR ERROR IF SET
1	OVF OVERFLOW/ UNDERFLOW	-DATA UNDERFLOW - PROGRAM SENT COLUMN VIA DATA INTERRUPT THAT INTERFACE COULD NOT ACCEPT. CARD WILL BE OFFSET IN HOPPER. OPERATOR SHOULD REMOVE CARD AND TELL SYSTEM TO RETRY.
2	PAR PARITY OR DATA CHECK	-PUNCH ERROR - HARDWARE ECHO CHECK. BAD CARD IS IN HOPPER AND OFFSET STACKED. OPERATOR SHOULD REMOVE CARD, READY DEVICE AND TELL SYSTEM TO RETRY.
3	INO INOPERABLE	-DEVICE DOES NOT EXIST. -JAM - TRANSPORT ERROR - POWER OFF - INTERLOCK OPEN. -WATCHDOG TIMER EXPIRED APPROX. 5 SEC. AFTER OPERATION INITIATED WITHOUT A DATA INTERRUPT OR APPROX. 5 SEC. HAS EXPIRED BETWEEN DATA INTERRUPTS. OPERATOR MUST OBSERVE ERROR CAUSE LIGHTS TO DETERMINE ERROR RECOVERY PROCEDURE.
4	MPE MEMORY PARITY	-NONE-
5	LOK WRITE LOCK	-NONE
6	CIG CHARACTER IN GAP	-NONE-
7	OTH OTHER	-NONE-
8	HOL HOLD	-NONE-

PROGRAMMER'S EVENT BIT IN UFT WORD UFTSTA	REASON CODE	REASON FOR EVENT IF SET
9	EOM END-OF- MEDIUM	-HOPPER EMPTY/STACKER FULL
10	EOF END-OF- FILE	-SET AFTER WEOF,REWIND,OR NULL OPERATIONS
11	BOM BEGINNING- OF-MEDIUM	-SET AFTER NULL OPERATIONS OR REWIND

Table 6-2. Card Punch Errors and Events

CHAPTER 7
MAX IV LINE PRINTER HANDLER (LPn,LCn,DPn)

The line printer handler consists of the standard three levels of execution, the TASK level and the directly connected routines for DATA and SERVICE interrupts utilizing hardware party-line interrupt vectors. These are permanent resident elements capable of controlling the transfer of data, status, and commands to/from a high-speed buffered line printer.

The hardware controller supports only one printer. However, additional printers may be added to additional controllers if each is assigned a unique group/unit number. A single copy of the handler will serve all of them.

The standard line printer has the controller name LP and uses group/unit #Ø7. The standard Direct Memory Processor (DMP) number is #7, if the printer is to be operated in this mode. Normally, the handler transmits data in a continuous burst at the service interrupt level. This interrupt level is entered as a result of the previous operation that was one or more commands to cause carriage motion operations. However, a system generation option may be specified (bit 2 of Logical Device Table word 4) that causes the handler to transmit bytes with the data interrupt level, at a slower 2Ø KB rate. The standard non-DMP logical device name is LP for the high-speed printer and LC for the matrix (Centronics) printer, which must be assigned to controller LP (or LC) at system generation.

The DMP or non-DMP mode is determined by specifying the appropriate controller name at system generation. The DMP controller name is DP; its standard logical device name is also DP, which must be assigned to the corresponding controller at system generation time. Additional printers must be assigned controller and logical device names of the form DPn, LCn, or LPn, where "n" is a unique character distinguishing each controller/device pair.

MAX IV includes a printer spooler, S. The reader is referred to the chapter in this manual on the MAX IV Imaginary Printer Spooler Symbiont.

NOTE: Because the multi-function controller is incapable of sustained burst mode operation, the DIO option must be selected when using the multi-function line printer controller or use the DP Handler. If it is not selected at SYSGEN time, the handler will put the printer offline and select the DI mode at run time.

7.1 I/O OPERATIONS AND MODES

7.1.1. STANDARD ASCII WRITE TO PRINTER (LPn,LCn,DPn)

This operation is identical to the teletypewriter WRITE operation. Carriage control bytes are interpreted in the Standard ASCII mode only. The first data byte in the buffer is interpreted by the handler and translated into a hardware command.

7.1.1.1 Character Set

The code set used by the printer is a truncated ASCII set of 64 characters; therefore, the non-printing control characters (for example CR, LF, BEL) should not be transmitted in Standard ASCII data buffers since the line printer truncates these codes to printable characters.

7.1.1.2 Carriage Control

The printer has a carriage control tape that may be used to specify non-standard print line positions on the page. Column 1 and column 8 of this tape must mark the top of form and end-of-form, respectively. The other columns may be punched in any desired form (double space, triple space, random). The end-of-form column should contain punches in all positions from the initial end-of-form line up to the top-of-form line on the next page. This ensures that the end-of-form status (EOF) may be sensed by the programmer if double- or triple-spaced lines fall anywhere in the end-of-page margin. This status bit is bit 10 of the UFT, the same as end-of-file for non-printing devices.

Since EOF is a status bit only returned to the programmer's User File Table (UFT) after printing any line in the end-of-page margin, the programmer may control paging operations at will. The programmer may choose to print the next line (and possibly a title) at the top of the next page ("-" or "1" carriage control bytes), when this status bit is sensed; or may choose to simply print throughout the margin region to conserve paper or to display continuous graphic information. MODCOMP standard software generally prints a standard title on each page when it senses end-of-form.

The top-of-form, single-space, double-space, and overprint carriage control options are performed with the standard ASCII carriage control bytes. The use of other carriage tape functions requires that the carriage control byte be a non-ASCII byte (most significant bit (0) set).

0	1	2	3	4	5	6	7
1	X	M	M	M	N	N	N

The X bit, if reset, causes MMMNNN line feeds of the carriage to occur before the text is printed. If X is set, the carriage advances to the next punched hole on the carriage tape in column NNN before the text is printed. Where NNN =Ø, column 1 is used by default.

Data bytes that exceed the printer's carriage width (system generation option) are lost. The transmission of data may be terminated by the handler short of the carriage width if it encounters the NUL marker word in the programmer's data buffer.

The Centronics printers may or may not have hardware form-feed capability. If the printer does have this capability, it is indicated to the handler by specifying the CFF bit in the LCn logical device option word.

7.1.2 NON-STANDARD ASCII WRITE TO PRINTER (LPn,LCn,DPn)

The Non-Standard ASCII mode does not interpret carriage control, and it upspaces one line before printing text. It terminates transmission of data when it encounters a byte in the programmer's buffer that matches the byte in the UFT option word (normally NUL). If set in the option word, bit 8 defeats this feature.

7.1.3 BINARY WRITE TO PRINTER (LPn,LCn,DPn)

The binary modes are not implemented for this symbolic printing device, but they are reserved for graphic modes that might be used by a special printer.

7.1.4 FILE/RECORD OPERATIONS

7.1.4.1 REWIND and HOME to Printer (LPn,LCn,DPn)

The REWIND and HOME operations cause the printer carriage to skip to the next hole in column 1 of the carriage tape (top-of-form) if the FF (form feed) bit is set in the Logical Device Table. If the FF bit is not set, the handler outputs four line feeds. In the DMP mode the device is assumed to have form feed capability and the FF bit is not tested in the Logical Device Table. REWIND additionally sets the device position index to zero in the file and sets the BOM and EOF event bits in the programmer's UFT.

7.1.4.2 ADVANCE RECORD to Printer (LPn,LCn,DPn)

The ADVANCE RECORD operation causes the printer carriage to advance one line. If the end-of-form column of the carriage tape has a hole punched in the new line position, the EOF event bit is set in the programmer's UFT status word.

7.1.4.3 ADVANCE FILE to Printer (LPn,LCn,DPn)

The ADVANCE FILE Operation causes the printer carriage to advance to the next hole in column 8 of the carriage tape (to end-of-form). The EOF event bit is set in the programmer's UFT if the carriage is left at the end-of-form margin.

7.1.4.4 WEOF to Printer (LPn,LCn,DPn)

This operation causes the printer to advance to end-of-form (carriage control column 8) and then prints the message "$$". The EOF event bit is set in the programmer's UFT. The operator may separate jobs easily when removing forms from the printer if this operation is transmitted at the end of each job. For "spooled" printers, this command defines the end of a spooled print file.

7.1.4.5 Null Operations to Printer (LPn,LCn,DPn)

The READ and BACKSPACE operations are not implemented for the line printer and perform as null operations if attempted. The EOF and BOM event bits are set in the programmer's UFT.

7.1.4.6 Summary Tables

Line Printer Handler summary tables appear at the end of this chapter. Table 7-1 contains information about the handler's response to I/O operations in the four modes of data transfer. Table 7-2 lists the error and event bits in the UFT status word (UFTSTA). Conditions causing the error or event are also listed.

7.1.5 DEFERRED PRINTING

The direct carriage movement operations (HOME, REWIND, WEOF, ADVANCE RECORD, and ADVANCE FILE) should be avoided in a program where the printed data is assigned directly to a non-printing secondary storage device. This type of logging is often done when the printing device is slow, so that the log data can be printed later by another off-line program. In this case, carriage control information should be passed only with WRITE commands, so that the data containing carriage control bytes inherently controls all carriage positioning. This restriction does not need to be imposed for a symbiont spooled device because the symbiont task can convert the direct carriage movement commands into dummy write operations to secondary storage.

7.2 ERROR RECOVERY AND HOLDUP CONDITIONS

The line printer, like all other handlers, allows all errors to be reported directly to the programmer or to the system operator. No error retry procedures are implemented. The following error causes immediate termination of the operation:

Device inoperable (for example, power off)

This device should not be timed with a watchdog timer because the OFFLINE switch on the printer inhibits service interrupts.

A message reporting a holdup condition is output to the operator in the following format:

```
!LPn?/tttttt/oooooo
```

tttttt - is the calling task's name

oooooo - is the current task overlay name

This message reports the following conditions:

- Device does not exist
- Device in hold state because the operator pressed the offline button
- Device under exclusive use by another task
- Device already offline

7.3 PRINTER QUALIFICATION

The printer qualification feature will permit page form control of those print devices (for example, DECWRITER) that are incapable of providing an EOF indication. The page length of any non-spooled print device that is not a MAXNET device may be set/varied at SYSGEN (DEVICE statement) or by an OC directive (/LEN, /WID).

Word 7 of the device LDT (LDTTOF) (Word 11 (LDTCLP) if connected to a communications PDT) maintains an end-of-page length and a present line counter adjusted by each print line's carriage control character. Special carriage control characters will advance the present line counter by one; single or multiple spaces will advance the count by that number of lines; a TOF command will reset the present line counter to one; a BOF or condition of present line count greater than or equal to end-of-page length will set EOF in the associated UFT. It is the user task's responsibility to check the EOF and issue the appropriate form command.

7.4 SYSTEM GENERATION

7.4.1 STANDARD PARAMETERS

The following statements are required in a system generation source file if the device is to be included in a system. Detailed descriptions of these statements and the rules of order in which they must be entered are included in the MAX IV SYSGEN, System Guide Manual.

```
CONTROLLER      DPn,7,7          for DMP LINE PRINTER device
          or
CONTROLLER      LPn,7            for LINE PRINTER device
          or
CONTROLLER      LCn,#B           for CENTRONICS device
     .
     :
```

```
DEVICE      DP,DPn,134      for DMP LINE PRINTER device
        or
DEVICE      LP,LPn,134      for LINE PRINTER device
        or
DEVICE      LC,LCn,134      for CENTRONICS device
```

7.4.2 DEVICE OPTION PARAMETER

The MAX IV Line Printer Handler is designed to function with different types of devices. However, some of these devices have peculiar characteristics that must be distinguished by the handler. The following bits may be set by parameter 4 of the DEVICE system generation statement to identify the device and its special function or characteristic.

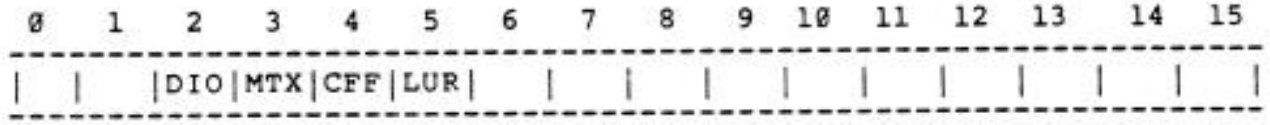


Figure 7-1. Line Printer Handler Device-Dependent Bits

Bit	Name	Meaning if Set (=1)
2	DIO	Indicates printing device is unbuffered and a data interrupt must be received prior to printing the next character. This bit is reset if parameter 4 is not specified.*
3	MTX	Indicates a Centronics-type printer which does not have normal carriage control capability (for example double space, skip to channel). If LC is the device name and parameter 4 is not specified, this bit is set by default.
4	CFF	Indicates the device has a hardware form-feed capability. This bit is examined by the handler only if MTX is set and is the default.
5	LUR	Indicates that lowercase characters (a-z) are to be converted to uppercase characters before printing. This is a software function and should only be included for uppercase only printers. (LINE PRINTER device, LP.HAN only)

* **NOTE:** Because the Multifunction controller is incapable of sustaining burst mode operation, the DIO option must be selected when using the multifunction line printer controller. If it is not selected at SYSGEN time the handler will put the printer offline and select the DI mode at run time.

LINE PRINTERS (LPn)

RECORD SIZE ≤ 80 BYTES OR ≤ 134 BYTES

	ASCII MODES		BINARY MODES	
	STANDARD	NON-STANDARD	STANDARD	NON-STANDARD
0 READ	null+	<------SAME	UNIMPLEMENTED	UNIMPLEMENTED
1 WRITE	OUTPUT ONE LINE TO PRINTER. OUTPUT CARRIAGE SKIP PREFIX AS PER CARRIAGE-CONTROL BYTE INTERPRETATION	OUTPUT 1 LINE TO PRINTER. UPSPACE BEFORE PRINT, TERMINATE ON BYTE IN UFT OPTION	UNIMPLEMENTED	UNIMPLEMENTED
2 REWIND	ADVANCE CARRIAGE TO TOP-OF-FORM (FORM FEED) COLUMN OF CARRIAGE CONTROL TAPE + *	<------SAME	UNIMPLEMENTED	UNIMPLEMENTED
3 BKFILE	null+	<------SAME	UNIMPLEMENTED	UNIMPLEMENTED
4 BKRECORD	null+	<------SAME	UNIMPLEMENTED	UNIMPLEMENTED
5 AVRECORD	ADVANCE CARRIAGE OF PRINTER ONE LINE	<------SAME	UNIMPLEMENTED	UNIMPLEMENTED
6 AVFILE	ADVANCE CARRIAGE TO BOTTOM-OF-FORM COLUMN OF CARRIAGE CONTROL TAPE. SET EOF EVENT BIT IN UFT. *	<------SAME	UNIMPLEMENTED	UNIMPLEMENTED
7 WEOF	ADVANCE CARRIAGE TO BOTTOM-OF-FORM (EOF) COLUMN OF TAPE, AND PRINT ($$) MESSAGE *	<------SAME	UNIMPLEMENTED	UNIMPLEMENTED
8 HOME	ADVANCE CARRIAGE TO TOP-OF-FORM (FORM FEED) COLUMN OF CARRIAGE CONTROL TAPE	<------SAME	UNIMPLEMENTED	UNIMPLEMENTED

* For Centronics only, top of form is defined by carriage control tape.
+ This operation sets the EOF and BOM event bits in the UFT status word.

Table 7-1. Line Printer Operations and Modes

PROGRAMMER'S ERROR BIT IN UFT WORD UFTSTA	OPERATOR'S REASON CODE	REASON FOR ERROR IF SET
1	OVP OVERFLOW/ UNDERFLOW	-NONE-
2	PAR PARITY OR DATA CHECK	-NONE-
3	INO INOPERABLE	-DEVICE DOES NOT EXIST -DEVICE INOPERABLE (POWER OFF)
4	MPE MEMORY PARITY	-MEMORY PARITY DETECTED (DMP MODE ONLY)
5	LOK WRITE LOCK	-PAPER IS LOW
6	CIG CHARACTER IN GAP	-NONE-
7	OTH OTHER	-NONE-
8	HOL HOLD	-NONE-
PROGRAMMER'S EVENT BIT IN UFT WORD UFTSTA	**REASON CODE**	**REASON FOR EVENT IF SET**
9	EOM END-OF- MEDIUM	-PAPER SUPPLY EXPIRED - OPERATOR SHOULD PUT DEVICE OFFLINE, REPLACE PAPER SUPPLY AND PUT DEVICE ONLINE.
10	EOF END-OF- FILE	-CARRIAGE PRINTED LAST LINE IN END-OF-FORM MARGIN -SET AFTER REWIND OR NULL OPERATIONS
11	BOM BEGINNING- OF-MEDIUM	-SET AFTER REWIND OR NULL OPERATIONS

Table 7-2. Line Printer Errors and Events

CHAPTER 8
MAX IV RASTER PRINTER/PLOTTER HANDLER (VPn)

The Versatec printer/plotter consists of the TASK level and directly connected DATA and SERVICE interrupt routines. The DATA interrupt is used for Direct Memory Processor (DMP) END-OF-BLOCK detection for completion of DMP transfers. The DATA and SERVICE interrupts utilize unique hardware party-line interrupt vectors. These are permanent resident elements so that they are always available without delay. These elements control the transfer of data, status, and commands to/from the raster plotter/printer device.

The hardware controller supports only one plotter/printer. However, additional plotter/printer devices may be added to additional controllers if each is assigned a unique group/unit number. A single copy of the handler will serve all of them.

The standard plotter/printer has the controller name VP and uses group/unit #Ø8. Because of a shared group/unit assignment, a system cannot have a Versatec Printer/Plotter and an Incremental X-Y Plotter in its hardware configuration with standard unit assignments.

DMP (#Ø7) is used because of the high data rate of the device. The data interrupt is thus used as an end-of-block interrupt for the DMP. The standard logical device name is VP, which must be assigned controller and logical device names of the form VPn, where n is a unique character distinguishing each controller/device pair.

The device can be used with continuous form rolls or 8-1/2" x 11" folded forms with the long dimension along the carriage. A hardware switch enables or disables the form-feed capability. The device does not have a carriage tape for carriage control nor does it provide a bottom-of-page status return to the programmer. Going to top-of-form is automatic when enabled and fanfold paper is being used. This causes the form-feed operation to top-of-form after a finite number of lines has been printed. If the device is spooled, the spooler can simulate the "end-of-form" status since the device does not provide this status.

8.1 I/O OPERATIONS AND MODES

ASCII operations control printing; Binary operations control plotting.

8.1.1 STANDARD ASCII WRITE TO PRINTER (VPn)

This Standard ASCII WRITE operation is similar to teletypewriter printer and line printer WRITE operations. Carriage control bytes are interpreted in the Standard ASCII mode only.

8.1.1.1 Carriage Control

Carriage control is effected by prefixing control bytes to the programmer's text instead of to actual carriage control specification bytes. This operation is accomplished by copying the programmer's buffer into a scratch buffer and prefixing the necessary carriage control bytes.

8.1.1.2 Character Set

Some versions of the printer accept the entire ASCII set; other versions print only a smaller set (not including lower case). Control characters are not printed.

8.1.1.3 Record Length

The printer accepts up to 132 characters in one unit record, including carriage control bytes. Less than a line can be specified by supplying a smaller buffer or by imbedding a null marker (two NUL bytes on a word boundary) in the buffer after the end of useful information. Data bytes in the programmer's buffer that exceed the carriage width are lost.

8.1.1.4 Print Line Operation

When a line is printed, the hardware actually plots a row at a time of the 7x9 dot matrix of each character on the line. A stepping motor moves the page carriage only 0.01 inch for each plot line taking 15 raster scans and steps to perform a complete print line operation (including margin between print lines).

The result of this motion is that the printer is always upspaced after print not requiring the normal upspace before print. The only disadvantage is that over-print operations cannot be performed. The top-of-form, single-space, and double-space operations are similar to other printing devices. No carriage control function is performed if the carriage control byte has its most significant bit set, since no hardware carriage skip feature is available.

8.1.2 NON-STANDARD ASCII WRITE TO PRINTER (VPn)

The Non-Standard ASCII mode does not interpret carriage control bytes. However, all bytes in the buffer are transmitted until a NUL marker word is found. This handler does not have the ability to stop transmission when a byte is found in the programmer's buffer that matches a byte in the UFT.

8.1.3 BINARY WRITE TO PLOTTER (VPn)

The Standard Binary mode is not implemented, but the Non-Standard Binary mode transmits raw data bits directly to the plotter "nibs" in a single line raster. The raster line consists of a number of light or dark spots across the page. Once the raster line is printed, the plotter automatically upspaces 0.01 inch. The raster line consists of black dots where "1's" exist in the bit string and no printed dot where "0's" exist. The dot positions, or nibs, are on 0.01 inch centers.

Generating the binary data for a raster plotter requires an entirely different software technique than is used to generate incremental plotter data. This is because the raster plotter advances paper in one continuous direction down a page while scanning across the pages. All data to be plotted spatially on a page must be produced by software scanning of the plot's "mathematical model" or by the use of random access storage of the plot data before it is transmitted to the raster plotter.

8.1.4 FILE/RECORD OPERATIONS

8.1.4.1 REWIND to Plotter/Printer (VPn)

The REWIND operation causes a carriage advance to the top-of-form position of the next page if the hardware form-feed switch is enabled. If this switch is disabled, a paper advance of approximately 8 inches occurs. The device position index in the UFT and file entry are reset to zero, and EOF and BOM event bits are set in the UFT status word.

8.1.4.2 ADVANCE RECORD to Plotter/Printer (VPn)

The ADVANCE RECORD operation causes the carriage to advance one line. If the form-feed switch is enabled and the hardware line count indicates the required number of lines has already been printed on the current page, an advance to top-of-form occurs automatically.

8.1.4.3 ADVANCE FILE to Plotter/Printer (VPn)

The ADVANCE FILE operation is functionally similar to REWIND except that the device position index is not reset and no event bits are set in the programmer's UFT.

8.1.4.4 WEOF to Plotter/Printer (VPn)

This operation upspaces the printer carriage two lines (if not already into the page margin) and then prints a "$$" message. The EOF event bit is set in the programmer's UFT. Following the printed line, the carriage is upspaced approximately eight inches. The purpose of this function is to remove the most recently printed lines from the developer bath at the end of a printing job so that they are not unnecessarily dark.

If device-independent programming is not required, an EOT character (end-of-transmission) imbedded at the end of the data buffer also causes a similar eight-inch upspace function to occur.

8.1.4.5 HOME to Plotter/Printer (VPn)

The HOME operation is functionally similar to the ADVANCE FILE operation, causing a carriage form-feed or upspace of eight inches.

8.1.4.6 Null Operations to Plotter/Printer (VPn)

The READ and BACKSPACE operations are not implemented and cause a null operation to occur, which sets EOF and BOM event bits in the programmer's UFT.

8.1.4.7 Summary Tables

Raster Printer/Plotter Handler summary tables appear at the end of this chapter. Table 8-1 contains information about the handler's response to I/O operations in the four modes of data transfer. Table 8-2 lists the error and event bits in the UFT status word (UFTSTA). Conditions causing the error or event are also listed.

8.1.5 DEFERRED PRINTING

The direct carriage movement operations (HOME, REWIND, WEOF, ADVANCE RECORD, and ADVANCE FILE) should be avoided in a program where the printed data is assigned directly to a non-printing secondary storage device. This type of logging is often done when the printing device is slow, so that the log data can be printed later by another offline program. In this case, carriage control information should be passed only with WRITE commands, so that the data itself (containing carriage control bytes) inherently controls all carriage positioning. This restriction does not need to be imposed for a symbiont spooled device because the symbiont task can CONVERT the direct carriage movement commands into dummy write operations to secondary storage.

8.2 UFT OPTIONS

The user indicates the data format mode and various options selected by setting bits in the UFT option word.

8.2.1 SIMULTANEOUS PRINT/PLOT CAPABILITY

The simultaneous print/plot feature requires optional hardware. If this hardware is present, it may be used in the following manner. Bit 1 of the UFT option word, if set, indicates the desire to use the simultaneous mode. To use this mode, the printed line must first be transmitted to the hardware character buffer. If Bit 1 in the UFT option word is set, the information is transmitted in the normal ASCII mode but is not actually printed. This operation must then be followed by 15 plot raster lines transmitted in the Non-Standard Binary mode but also with Bit 1 set. This binary transmission can be in one WRITE operation or in 15 separate WRITE operations. As each raster line is transmitted, it is plotted but each row of the print text in the hardware character buffer is logically "OR'ed" with the plot data.

8.3 ERROR RECOVERY AND HOLDUP CONDITIONS

The printer/plotter handler, like all other handlers, allows all errors to be reported directly to the system (operator) or to the programmer. No error retry procedures are implemented. The following errors cause immediate termination of the operation.

- Device inoperable (for example, power off).
- Watchdog timer expired (also sets device inoperable status) approximately 10 seconds after last operation initiated (optional).

A message reporting a holdup condition is output to the operator in the following format:

```
!VPn?/tttttt/oooooo
```

tttttt - is the calling task's name

oooooo - is the current task overlay name

This message reports the following conditions:

- Device does not exist
- Device under exclusive use by another task
- Device already offline

8.4 SYSTEM GENERATION

The following statements are required in a system generation source file if this device is to be included in a system. Detailed descriptions of these statements and the rules of order in which they must be entered are included in the MAX IV SYSGEN, System Guide Manual.

```
CONTROLLER                  VPn,8,7,,,,67
    .
    .
    .
DEVICE                      VP,VPn,134
```

Parameter 7 of the CONTROLLER statement specifies the size, in words, of the scratch buffer to be used by the handler for Standard ASCII writes. The buffer size should equal the number of characters per print line +2.

Parameter 3 of the DEVICE statement specifies the natural block size in bytes for binary writes (plotting). The minimum block size needed to "plot" one line across the page can be calculated from printer specifications. A nib correlates to one bit of data. Block size calculations use the following formula:

$$\text{maximum plotting width in inches} \times \text{linear nib density in nibs per inch} \times \text{1 byte per 8 bits}$$

EXAMPLES

Printer specifications show that the Model 1100A has a 10.24 inch maximum plot width and a linear nib density of 100 nibs per inch. To convert to a buffer size:

$$10.24 \text{ inches} \times \frac{100 \text{ nibs}}{\text{inch}} \times \frac{\text{byte}}{8 \text{ bits}} = \frac{1024}{8} = 128 \text{ bytes}$$

```
DEVICE      VP,VPn,128
```

The Model 1200 has a 10.56 inch maximum plotting width and a linear nib density of 200 nibs per inch. To convert to a buffer size:

$$10.56 \text{ inches} \times \frac{200 \text{ nibs}}{\text{inch}} \times \frac{\text{byte}}{8 \text{ bits}} = \frac{2112}{8} = 264 \text{ bytes}$$

```
DEVICE      VP,VPn,264
```

	RECORD SIZE PLOTTER ≤ 32768 BYTES		PRINTER ≤ 132 BYTES	
	<----------ASCII MODES---------->		<----------BINARY MODES---------->	
	STANDARD	NON-STANDARD	STANDARD	NON-STANDARD
0 READ	null+	<-------SAME	<-------SAME	<-------SAME
1 WRITE	OUTPUT ONE LINE TO PRINTER. OUTPUT CARRIAGE SKIP PREFIX AS PER CARRIAGE CONTROL BYTE INTERPRETATION *NON-SIMUL-TANEOUS PLOT	OUTPUT ONE LINE TO PRINTER. CARRIAGE WILL BE UPSPACED AUTOMATICALLY. *NON-SIMUL-TANEOUS PLOT	UNIMPLEMENTED	TRANSMIT BIT RASTER LINE OR BLOCK (128 BYTES = 1024 BITS = ONE LINE) *NON-SIMUL-TANEOUS PLOT
2 REWIND	ADVANCE CARRIAGE TO TOP-OF-FORM (FORM FEED) POSITION OR SKIP 8" (SWITCH SELECTABLE)+	<-------SAME	<-------SAME	<-------SAME
3 BKFILE	null+	<-------SAME	<-------SAME	<-------SAME
4 BKRECORD	null+	<-------SAME	<-------SAME	<-------SAME
5 AVRECORD	ADVANCE CARRIAGE OF PRINTER ONE LINE(IF NOT PAST BOTTOM-OF-FORM).	<-------SAME	<-------SAME	<-------SAME
6 AVFILE	ADVANCE CARRIAGE TO TOP-OF-FORM (FORM FEED) POSITION OF SKIP 8" (SWITCH SELECTABLE)	<-------SAME	<-------SAME	<-------SAME
7 WEOF	PRINT ($$) MESSAGE WITH CR,2LF LEADER PREFIXED. 8" PAPER TRAILER (EOT) FOLLOWS	<-------SAME	<-------SAME	<-------SAME
8 HOME	ADVANCE CARRIAGE TO TOP-OF-FORM (FORM FEED) POSITION OR SKIP 8" (SWITCH SEL.)	<-------SAME	<-------SAME	<-------SAME

* This function defeated if Bit 1 of UFT option word set.
+ This operation sets the EOF and BOM event bits in the UFT status word.

Table 8-1. Raster Printer/Plotter Operations and Modes

PROGRAMMER'S ERROR BIT IN UFT WORD UFTSTA	OPERATOR'S REASON CODE	REASON FOR ERROR IF SET
1	OVF OVERFLOW/ UNDERFLOW	-NONE-
2	PAR PARITY OR DATA CHECK	-NONE-
3	INO INOPERABLE	-DEVICE DOES NOT EXIST -DEVICE POWER OFF -WATCHDOG TIMEOUT APPROX. 10 SEC AFTER OPERATION INITIATION
4	MPE MEMORY PARITY OR IOP ERROR	-MEMORY PARITY ERROR WHILE DMP WAS TRANSFERRING DATA TO PLOTTER/PRINTER -INVALID BUFFER ARGUMENTS
5	LOK WRITE LOCK	-NONE-
6	CIG CHARACTER IN GAP	-NONE-
7	OTH OTHER	-NONE-
8	HOL HOLD	-NONE-
PROGRAMMER'S EVENT BIT IN UFT WORD UFTSTA	**REASON CODE**	**REASON FOR EVENT IF SET**
9	EOM END-OF-MEDIUM	-PAPER SUPPLY HAS EXPIRED
10	EOF END-OF-FILE	-SET AFTER WEOF OR REWIND OR NULL OPERATIONS
11	BOM BEGINNING-OF-MEDIUM	-SET AFTER REWIND OR NULL OPERATIONS

Table 8-2. Raster Printer/Plotter Errors and Events

CHAPTER 9
MAX IV INCREMENTAL X-Y PLOTTER HANDLER (XYn)

The incremental plotter handler consists of directly connected routines for SERVICE and DATA interrupts utilizing hardware party-line interrupt vectors. These are permanent resident elements so that they are always available without delay. These elements control the transfer of data, status, and commands to/from the incremental plotting device.

The hardware controller supports only one plotter. However, additional plotters may be added to additional controllers if each is assigned a unique group/unit number. A single copy of the handler will serve all of them.

The standard plotter has the controller name XY and uses group/unit #08. Because of a shared group/unit assignment a system cannot have a Versatec Plotter/Printer and an Incremental X-Y Plotter in its hardware configuration with standard unit assignments. A Direct Memory Processor (DMP) is not required since the data rate is low enough to be supported by data interrupts. The standard logical device name is XY, which must be assigned to controller XY at system generation. Additional plotters must be assigned controller and logical device names of the form XYn, where n is a unique character distinguishing each controller/device pair.

9.1 I/O OPERATIONS AND MODES

9.1.1 ASCII AND STANDARD BINARY

The ASCII format and the Standard Binary format are not defined for this device. Only a Non-Standard Binary form of data transmission is recognized.

9.1.2 BINARY WRITE TO PLOTTER (XYn)

This Binary WRITE operation only has definition in the Non-Standard mode. A series of binary-coded plotter commands are transmitted. A unit record size is defined directly by the size of the programmer's data buffer. Each 8-bit byte in this buffer transmits 6 bits of control information to the plotter. These 6-bit commands are unitary in definition. Refer to Figure 9-1.

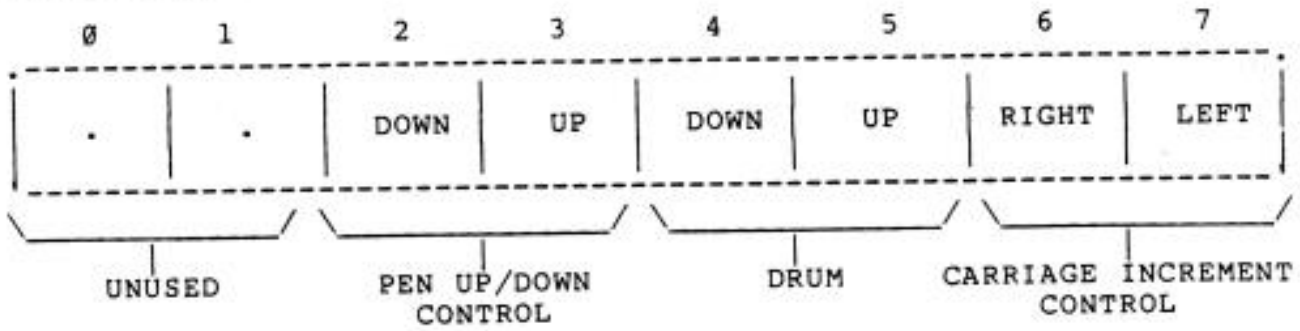


Figure 9-1. Plotter Control Information

If any bit is set, the corresponding operation is performed. Note that the operations are organized in pairs of bits. Each pair is mutually exclusive with other pairs in that each pair can have one bit set. This allows motion of the pen, drum, and carriage to be specified as one "simultaneous" operation. Setting both bits in a pair is not allowed, for obvious reasons. When pen motion is specified in the same command with drum/carriage motion, the slower pen command is performed first.

Since only an even number of bytes, packed two per word, can be transmitted in one unit record, a null (all bits zero) byte may be transmitted at any time to pad out to a word boundary or for timing purposes.

Notice that all plotting operations are single increments relative to the previous position of the pen. This usually requires that the programmer use a "coordinate" system as a frame of reference. The software necessary to support coordinate plotting is not part of the handler itself. Such a software package is available from MODCOMP, however.

9.1.3 FILE/RECORD OPERATIONS

9.1.3.1 REWIND to Plotter (XYn)

The REWIND Operation causes the logical device (XYn) to be put into an "off-line" condition that is reported directly to the operator. When a REWIND operation is performed, all operations already in the controller's queue are dequeued into the slough queue of the logical device. No further operations can be queued to this plotter until the operator intervenes by putting the device back into the "on-line" state. At that time, all waiting operations are requeued in the controller's queue. The reason code for the off line condition is "HOL" (see Errors and Events, Table 9-2). Normally, this operation should be performed at the beginning of a new plot because it is possible the operator may have moved the origin when removing the previous plot. The REWIND operation allows a programmer to inform the operator whenever it is necessary to synchronize the plotter's physical origin to the coordinate system origin being used by the programmer. Because of the incremental nature of the device, no absolute origin can be set by the program itself.

The response required by the operator to put the device into the on-line state is:

```
/ON XYn NO
```

where "NO" indicates "no retry", and "n" describes the particular plotter and may be missing if there is only one plotter. The device position index is reset to zero and the BOM and EOF event bits are set in the User File Table (UFT) status word.

9.1.3.2 WEOF to Plotter (XYn)

The WEOF operation is similar to REWIND except the reason code is "OTH" (refer to Table 9-2). This operation should be performed when plotting operations are complete so the operator knows to remove the plot. The EOF event bit is set in the UFT status word and the device position index is incremented.

9.1.3.3 HOME to Plotter (XYn)

The HOME operation is identical to REWIND except that the device position index is not modified and no event bits are set in the UFT status word.

9.1.3.4 Null Operations to Plotter (XYn)

The READ, BACKSPACE, and ADVANCE operations perform null operations. The EOF and BOM event bits are set in the programmer's UFT.

9.1.3.5 Summary Tables

X-Y Plotter Handler summary tables appear at the end of this chapter. Table 9-1 contains information about the handler's response to I/O operations in the four modes of data transfer. Table 9-2 lists the error and event bits in the UFT status word (UFTSTA). Conditions causing the error or event are also listed.

9.2 HOLDUP CONDITIONS REPORTED TO OPERATOR

A message reporting a holdup condition is output to the operator in the following format:

```
!XYn ?/tttttt/oooooo

     tttttt - is the calling task's name

     oooooo - is the current task overlay name.
```

The message reports the following conditions:

- Device does not exist
- Device under exclusive use by another task
- Device already offline.

9.3 SYSTEM GENERATION

The following statements are required in a system generation source file if this device is to be included in a system. Detailed descriptions of these statements and the rules of order in which they must be entered are included in the MAX IV SYSGEN, System Guide Manual.

```
CONTROLLER                    XYn,8
     .
     .
     .
DEVICE                        XY,XYn
```

RECORD SIZE ≤ 32768 BYTES

	ASCII MODES		BINARY MODES	
	STANDARD	NON-STANDARD	STANDARD	NON-STANDARD
0 READ	SAME-------->	SAME-------->	SAME-------->	null+
1 WRITE	UNIMPLEMENTED	UNIMPLEMENTED	UNIMPLEMENTED	TRANSMIT PEN CONTROL BYTES IN RECORD = SIZE OF USER'S BUFFER. (6 BIT UNITARY BYTES RIGHT JUSTIFIED IN 8 BIT BYTE)
2 REWIND	SAME-------->	SAME-------->	SAME-------->	PUT XY PLOTTER DEVICE OFFLINE WITH REASON CODE "HOL" (LETS OPERATOR SET ORIGIN)
3 BKFILE	SAME-------->	SAME-------->	SAME-------->	null+
4 BKRECORD	SAME-------->	SAME-------->	SAME-------->	null+
5 AVRECORD	SAME-------->	SAME-------->	SAME-------->	null+
6 AVFILE	SAME-------->	SAME-------->	SAME-------->	null+
7 WEOF	SAME-------->	SAME-------->	SAME-------->	PUT XY PLOTTER DEVICE OFFLINE FOR REASON CODE "OTH" (LETS OPERATOR REMOVE PLOT)
8 HOME	SAME-------->	SAME-------->	SAME-------->	PUT XY PLOTTER DEVICE OFFLINE FOR REASON CODE "HOL" (LETS OPERATOR SET ORIGIN)

+ This operation sets the EOF and BOM event bits in the UPT status word.

Table 9-1. Incremental X-Y Plotter Operations and Modes

PROGRAMMER'S ERROR BIT IN UFT WORD UFTSTA	OPERATOR'S REASON CODE	REASON FOR ERROR IF SET
1	OVF OVERFLOW/ UNDERFLOW	-NONE-
2	PAR PARITY OR DATA CHECK	-NONE-
3	INO INOPERABLE	-POWER OFF OR OFFLINE - OR DOES NOT EXIST
4	MPE MEMORY PARITY	-NONE-
5	LOK WRITE LOCK	-NONE-
6	CIG CHARACTER IN GAP	-NONE-
7	OTH OTHER	-WEOF OPERATION HAS BEEN PERFORMED. PUT DEVICE BACK ONLINE AFTER REMOVING PLOT(S) AND RESETTING ORIGIN FOR NEXT PLOT.
8	HOL HOLD	-REWIND OR HOME OPERATION HAS BEEN PERFORMED. PUT DEVICE BACK ONLINE AFTER RESETTING ORIGIN FOR NEXT PLOT.
PROGRAMMER'S EVENT BIT IN UFT WORD UFTSTA	**REASON CODE**	**REASON FOR EVENT IF SET**
9	EOM END-OF-MEDIUM	-NONE-
10	EOF END-OF-FILE	-SET AFTER WEOF, REWIND, OR NULL OPERATIONS
11	BOM BEGINNING-OF-MEDIUM	-SET AFTER REWIND OR NULL OPERATIONS

Table 9-2. Incremental X-Y Plotter Errors and Events

CHAPTER 10
MAX IV IMAGINARY PRINTER SPOOLING SYMBIONT

Spooling is defined here as the interposing of some form of high-speed buffering between a task program trying to print and the actual printing device. Spooling is accomplished by a symbiont task that intercepts and processes the requests from several imaginary printers, which appear to be standard MODCOMP printer devices.

The spooler eventually funnels the disc-resident print data to one or more real printers. This spooler may be used with the output channel of a slow full-duplex or half-duplex terminal device printer or with high-speed line printers. When a real printer is assigned the job of printing the data from more than one imaginary printer, only one print file is chosen for output. The other jobs are stored and are printed sequentially.

10.1 SPOOLING AND THE PROGRAMMER

To print using a spooled printing device, the programmer directs all print data (unit lines) to a logical file. This file need only be assigned to an imaginary spooling device instead of a direct printing device. The programmer can address this print file with WRITE, REWIND, AVRECORD, AVFILE, WEOF, HOME, and TERMINATE service calls. The WRITE service may pass packed ASCII bytes (two per word) in either the Standard ASCII mode or the Non-Standard ASCII mode. Binary modes are not supported. The WEOF service is used to define the limits of print files.

The implementation of spooling for a device does not block its usage as a direct device. Any task program can choose to use the appropriate direct or spooled print path by assigning the print file to the true device or the imaginary spooled device. However, the true printer can be assigned exclusively to the spooling task if this restriction is desired.

10.2 SPOOLING SYMBIONT

Spooling is controlled by a single resident or prescheduled task "S" added to the system during system generation. The S task consists of two parts.

The first part is called the Imaginary Device Section (IDS). Its purpose is to accept all printing requests from calling task programs using one or more imaginary print devices and to simulate actual "fast" printing devices(s) by writing the print data on a disc partition. The second part is called the Physical Device Section (PDS). Its purpose is to print on an actual printer the data stored on disc by the first part of the symbiont. The task considers IDS to be a "higher priority" section than PDS.

As soon as the Imaginary Device Section has "processed" a printed data record, it "fools" the calling task into believing that the line is printed quickly, even though the final target may be a slow printing device. As long as the print records come in bursts and buffering space remains available on the disc, the imaginary DEVICE continues to act as a fast printer. If output is sustained for long periods so that disc buffering space is demanded faster than the Physical Device Section can "drain" it to the physical printer, the specified amount of disc space will become saturated. When this happens, the imaginary device defaults to the speed of its target physical printer until disc space becomes available. The size of the buffering disc partition(s) can be adjusted to avoid such conditions in practical applications. Disc partitions can be sized appropriately for each spooling requirement.

10.3 IMAGINARY DEVICE SECTION (IDS) OPERATION

The IDS part of the spooling task contains the equivalent of a Physical Device Table (PDT) at the beginning of its task common (body) area. Any requests for operations to be performed on an imaginary printing device belonging to the task are queued by the I/O system in the PDT as operation nodes. The MAX IV I/O system queues an indefinite string of operation nodes whose positional order in the queue is determined first by calling task priority and then chronologically for like task priorities. The "S" task is "resumed" by the I/O system whenever the first operation node is placed in the queue and whenever that node is removed prematurely (for example, the operation is terminated or the calling task is aborted). The I/O system uses only the SHUTDOWN bit (0) and CONTROLLER BUSY bit (1) in the PDT status word to communicate with IDS. Other bits in this status word may be used internally by the symbiont. CONTROLLER BUSY is set as long as a node can be processed in the Physical Device Queue (PDQ) and must be reset by IDS whenever it finishes processing the last processable node in the queue. The SHUTDOWN bit is reset by the I/O system when the first node is placed in the PDQ and whenever it is prematurely removed. IDS must set this bit while processing a node. After the node has been processed, IDS must always test this bit to make sure it is still set before proceeding with processing of the next node. If it should be reset, IDS must assume the node has been removed and must initialize itself in such a manner that it "forgets" about the operation node it was working on.

IDS sloughs all operation nodes that it cannot process into the slough queue, which is kept in the Transport Control Table of the imaginary device. Reasons for sloughing an operation node include the device being off-line (manual operator directive OFF) or being under the exclusive use of a task other than the calling task. These nodes are subsequently requeued by the ON directive or by the "giving up" of exclusive use by the task that has it.

IDS plays the role of a "service interrupt", becoming active whenever it is triggered by the I/O system. Its primary goal is always to "perform" the operation represented by the first node in the queue. It then proceeds to other nodes that may exist in the queue whenever the first operation node is finished and returned to the calling task. When the queue is empty, or when no node can be processed because of lack of disc space, or when all internal core buffers are busy, IDS enters the PDS part of the spooling task.

When IDS finishes processing the first operation node in the queue, it returns it to the available node pool, resets the UFT entry in the node, and resumes that task if the original queued operation was a wait-mode operation. The active node counter is decremented in the Task Control Block (TCB) of the calling task. If no other nodes are found in the queue, IDS resets the CONTROLLER BUSY status bit and enters the PDS part of the spooling task.

The IDS part of the spooler contains, or has available to it, a "compression" buffer for each imaginary device it must simulate. Each buffer is equal in size to a disc sector of the spooling disc partition (usually 100 or 128 words). These buffers are used by IDS to move data from the ASCII print buffer of a task attempting to print on an imaginary spooling device. Any string of contiguous repeating characters (for example, blanks) in print lines are suppressed and converted into a pair of control bytes while the data is being moved into the currently used compression buffer. Other control bytes are used to indicate the end of a unit record (line) so that many short messages can be collected in the compression buffer before any disc write operation is attempted. When a full compression buffer is obtained, IDS moves the full compression buffer to a WRITE compression buffer and then queues a WRITE operation to disc while the compressing and moving of the print data continues. If both buffers are full, or if no more nodes are to be processed and at least one of the buffers is full, available unused sectors are allocated from the buffering disc partition. Any full compression buffers are written on a disc partition unique to each imaginary spooling device.

The IDS part of the spooler maintains tables within its body that permit it to find available disc sectors in a particular device's buffering partition and to determine when this space is exhausted. These tables are also available to the PDS part of the spooler so that it can return sectors to the "available-sector-list" when it finishes printing the data.

These sectors may be deallocated immediately upon PDS reading the sector, or deallocation may be deferred until a complete print file is printed (a number of print lines followed by a file mark).

The IDS part of the spooler maintains tables that permit it to build the disc sector files of compressed print data. Each imaginary print device has its own dedicated disc partition.

IDS may have a maximum of 16 DEVICES connected to it. These separate logical devices are given unique characteristics (for example, maximum byte count, printer/terminal) at system generation time that govern how IDS will spool and queue their data files and how PDS will print these files.

Each "emergency message" type DEVICE is usually specified on a dedicated printer (for example, the operator's terminal) so that the print information, which collects on the disc, can be printed on a first-come-first-served basis. Each "report type" DEVICE (which normally has a separate disc partition associated with it) acts like a physical device. If only one task outputs to a particular DEVICE, this user does not have any other user's output intermixed. However, if two or more tasks output to the same DEVICE, the output is intermixed on a first-come-first-served basis. Likewise, if a task outputs separate reports to two different DEVICES, the reports are separated even if they are printed on the same physical printer.

These devices can also have a "retain file" option specified that causes a completed print file queue to be retained. PDS does not print beyond one file mark until operator instructions are given with this option specified.

The spooler can run at any level the user desires, but it must run as a privileged task.

In addition to the WRITE operation, the "S" task responds to other I/O functions in a manner similar to an actual printing device:

- REWIND/ADVANCE FILE operations cause a top-of-form operation to be spooled on the target physical printer.
- ADVANCE RECORD causes a line to be upspaced.
- WEOF causes the end of the current print file to be communicated to PDS and tells PDS to switch to the next DEVICE of the same physical printer. It also causes the target physical printer to have a "$$" record printed at end-of-form (bottom of page).
- HOME operates the same as REWIND.
- READ and BACKSPACE FILE/RECORD are ignored operations.
- REWIND, HOME, ADVANCE FILE, ADVANCE RECORD, and WRITE operations communicate page boundaries to PDS, for purposes to be defined later.
- An effective page size is selectable by the operator for each DEVICE. The page size forces "end-of-form" indications to the calling task at boundaries appropriate to the physical printer to be used for the final printing.

10.4 PHYSICAL DEVICE SECTION OPERATION

PDS supports any combination of imaginary and real physical printing devices. If there is a one-to-one correspondence between imaginary and real devices, PDS attempts to keep each physical device busy printing the data passed to each corresponding imaginary device. If more than one "report type" of imaginary device is being spooled, PDS holds back the printed data of most of the similar imaginary devices while trying to keep the physical device busy printing a "file" of data for one of them. If more than one physical printer is available, additional print queues can be printed in parallel. Whenever a file mark ($$) is encountered, the currently serviced imaginary-device sector queue is stalled and another imaginary device is selected, until the bounds of its print file are detected (another $$ line). This rotating service pattern continues in a round-robin fashion until one print file per imaginary device is printed in turn.

PDS has one "uncompress" buffer for each physical device that it must service. This buffer is sized according to the size of the disc sector. When data is available to be printed on a particular physical device, this buffer is used to read the compressed ASCII from disc and to unpack the individual print lines or functions into a "line" buffer that is two bytes greater in size than the physical printer column width (one of these "line" buffers is also included within the body of the PDS for each physical device). When a full or partially full compressed print data buffer is obtained from IDS, PDS proceeds to print all lines contained within the "uncompress" buffer.

Normally, uncompression buffers are filled from the "oldest" unprinted disc sector in the "chosen" imaginary device queue. If PDS finds no data in the disc file of an "emergency message" DEVICE, it goes directly to the compress buffer in IDS for the selected device. If this buffer has been partially filled, the data is moved directly to the uncompress buffer of the physical device. Uncompression and printout are then initiated. When printing demands are low, this core-to-core buffering bypasses disc buffering and printout occurs in bursts. An option must be set for "report type" DEVICES not to bypass the disc in this manner.

When PDS finishes filling its uncompression buffer from the disc partition, it then institutes disc-space deallocation procedures before or during the uncompression and printing operations. If the uncompression buffer content were obtained directly from the IDS with a core-to-core move, PDS makes the source compression buffer "empty" and again available to IDS. If the buffer was obtained from a disc sector, PDS uses various methods of deallocating available sectors into the pool used by IDS.

If a certain option is specified (called the KEEP FILE option), PDS does not return each sector to the available pool when uncompression and printout starts, but it retains it in the imaginary device print queue until PDS finds a file boundary. In the case of retention, several pointers into such a queue are maintained. One points to the beginning of the first sector in the print file or page, and another points to the sector to be printed

next by PDS. A third pointer marks the last sector of printed information that has been spooled from the imaginary device by IDS. When an end-of-file record ($$) is encountered that originated from the calling task, the third pointer continues to build disc sectors for the next file of information. As PDS processes the print data strings for DEVICES that have the KEEP option specified, it does not deallocate the individual sectors already printed but retains the entire data file. Actually, sector deallocation is not done until the space is needed by IDS. Thus, the KEEP option guarantees that space is not deallocated until the operator specifies deallocation.

When an end-of-file record is encountered by PDS for a print queue that has "kept" an entire file, the printing of data for this imaginary device is suspended until an operator communication response is received.

In the meantime, other imaginary devices are given a chance to print their current files. The operator's response can be to "REMOVE" or to "REPEAT" the stalled file. If "REMOVE" is chosen, all sectors in the retained file are deallocated as needed. If "REPEAT" is chosen, PDS resets the "next-sector-to-be-printed" pointer for the imaginary device to the beginning of the retained file. Meanwhile, if other printing data is being spooled to this imaginary device, it continues to be strung into the other available parts of the file after the file mark as additional print files. The sector containing the file mark record is "run out" so that the next print file starts on a sector boundary. When PDS again chooses this imaginary device for printout to the physical device, the "KEPT" file is repeated up to the file mark. This repeat process is continued for this imaginary device as long as the operator requests "REPEAT". When the operator requests "REMOVE", all sectors in the retained file are deallocated as needed, and PDS proceeds to the next file of the next imaginary device (if several imaginary devices share a single physical printer), or it proceeds to the next print file of the current imaginary device (if no other imaginary device has data ready for a physical printer).

When PDS can find no full sectors to print in any imaginary device queue, it enters a relinquish period. After returning from the relinquish, if it still cannot find any print data on disc, it enters the core-to-core mode. If no print data is found in the core buffers or on disc, and there are no nodes for IDS to process, PDS suspends the "S" task and the task remains suspended until the I/O system places another node in the queue.

10.5 OPTIONAL DESIGN FEATURES

One double-compress buffer is shared by all DEVICES assigned to IDS. This buffer overlays the spooler initialization routine; thus, this space is essentially free.

The spooler permits any queued print file to be held or immediately deallocated by operator action. Other held print files can be initiated or resumed. If a print file is aborted in favor of another, it starts from the beginning when resumed.

10.6 INSTALLING THE PRINTER SPOOLER

10.6.1 TASK PREPARATION

The standard spooler has the name "S". If the spooler is to be made a normal resident task, a TASK statement must be specified in the Task Structure Block portion of the system generation. The spooler uses the initialization part of its code as a buffer; therefore, a resident spooler cannot be restarted or aborted without reloading the system.

A TASK statement for a resident spooler contains the following information:

```
                    1     2           3
TASK                S[,priority][,influence]
```

S - Parameter 1 is the spooler task name.

[priority] - Parameter 2 is the priority level at which the spooler is to run. This level must be equal to or higher than (a number closer to 0) the tasks that use it. Parameter 2 is optional and if not specified defaults to 127.

[influence] - Parameter 3 specifies the highest priority level at which this task is allowed to execute. This influence limit also determines the extent this task can influence and be influenced by other tasks.

Example:

```
TASK  S,100          A resident spooler is specified.
```

A non-resident spooler "S" should be avoided if the core memory requirements of the system are limited. The system will stall if the loading of the spooler causes the rolling of other tasks that may be using it.

If the spooler is to be a prescheduled task (at the expense of a little more core), the SYSGEN statement should read:

```
                          1      2         3
PRESCHEDULE               S[,priority][,file]
```

S	- Parameter 1 is the spooler task name.
[priority]	- Parameter 2 is the priority level at which the spooler is to run. This level must be equal to or higher than the priority of tasks that use it. Parameter 2 is optional and is superceded by the priority level used when cataloged.
[file]	- Parameter 3 is the name of the file (usually SM) from which the spooler is loaded.

Example:

```
PRESCHEDULE S,,SM          Priority level is cataloged with the
                           non-resident task.
```

To specify the CONTROLLER for the spooler task, use the following statement in the logical I/O portion of the SYSGEN in the form:

```
                    1    2
SYMCONTROLLER       S,[buffer]
```

S - Parameter 1 is the spooler task name.

buffer - Parameter 2 is optional. It specifies the size of the resident spooler's buffer. This parameter is not used by a non-resident (prequeued) spooler. The spooler uses this buffer to contain all of its local buffers. The size of this buffer must, therefore, be computed from a formula that depends upon the number of imaginary printer devices and the number of physical printers to be used by the spooler. The formula for computing the size of this buffer in words is:

$$buf = \sum_{i=0}^{15} (sectorsize) + \sum_{p=0}^{15} (sectorsize_{max}) + carriage/2 + 12$$

where: $(sectorsize_{max})$ is the largest of the (sectorsize).

sectorsize - is the sectorsize in words of the disc partition used to spool the imaginary device.

i - is the number of imaginary printer devices specified by DEVICE statements in the system generation deck for this task.

carriage - is the carriage width of the physical printers in bytes, including 2 bytes for carriage control. This value will also be specified in the DEVICE statement.

p - is the number of physical printers the spooler will use to distribute the printing load from all the imaginary devices.

If the sector sizes of disc partitions are the same for all imaginary print files and if the carriage width of all physical printers is the same, the above formula reduces to:

```
buffer = i * (sectorsize) + (sectorsize + 12 + carriage/2) *p
```

There must be a LOGFILE statement for every disc UFT number and for every physical printer UFT number that was specified by DEVICE statements for this spooler task. In addition, there has to be a LOGFILE statement for error and status communications between the spooler and the "system" operator (this logfile is called "OUT").

To specify which "imaginary" printer devices the spooler will simulate, use the following statement in the Logical I/O Block portion of the system generation for each such device:

```
DEVICE          ddn,S,bpr,dop
```

ddn - is the name of the imaginary device (usually LP1 through LPF).

S - is the name of the standard spooler task (if only one spooler is to be used).

bpr - is the maximum number of bytes per record. This is used by the spooler task and is used by various processors and the FORTRAN Run-Time Package to determine the maximum number of bytes per line. It should be the size of the carriage width (plus carriage control) of the physical printer that will be ultimately used (usually 72+2, 132+2, or 80+2).

dop - is a word that identifies the device and assigns physical printers and disc files to the imaginary device. The first four bits (0-3) of the word specify a "relative" physical printer number. The second four bits (4-7) specify a "relative" disc file number. The remaining eight bits (8-15) specify the device type (for example, printer). The spooler looks at the bits 10 and 13 of the last eight bits (8-15); however, most processors look at some of them. Therefore, at least the bit identifying a printer (#20) should be set. The first four bits (0-3) are used to form the print file UFT names in the range of P00-P15. If the first four bits are the same for more than one device and the second four bits are different, the spooler forms a round-robin queue of these task devices so that they share the physical printer. The second four bits (4-7) are used to form the DISC UFT names in the range of

D00-D15. The file names that are generated by the above assignments must also be used to define corresponding logical files in the spooler's RESOURCE BLOCK.

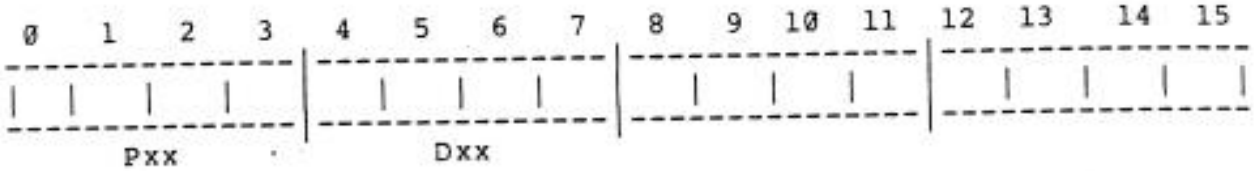


Figure 10-1. Imaginary Printer Device Option Word

Bits	Meaning if Set
0-3	Relative Physical printer number. Used to form the print file UFT names in the range P00 to P15. Note, the decimal value of these bits is used in the filename.
4-7	Relative disc file number used to form the disc UFT names in the range D00 to D15. Note, the decimal value of these bits is used in the filename.
10	Identifies the device as a printer.
13	Identifies the device as having a Keyboard.

Example 1:

```
DEVICE      TY1,S,74,#0024
```

Example 2:

```
DEVICE      LP1,S,134,#1420
```

Example 3:

```
DEVICE      LP2,S,134,#1220
```

Example 1 defines TY1 as an imaginary printer connected to task S. The maximum number of bytes that can be written to this device is 74. The spooler will write to the physical printer assigned by a UFT name of P00. The spooler will write and read to the disc partition assigned by a UFT of name D00. The last eight bits (8-15) of word "dop" (#24) specify that this is a printing terminal device. Files P00 and D00 must be defined and assigned in task S's resources.

Example 2 defines LP1 as an imaginary printer connected to task S. The maximum number of bytes that can be written is 134. The spooler will write to the physical printer assigned by a UFT of name P01. The spooler will write and read to the disc partition assigned by a UFT of name D04. The last eight bits (8-15) of word

"dop" (#20) specify that this is a printing device. Files P01 and D04 must be defined and assigned in task S's resources.

Examples 2 and 3 show how to specify two logical printers that will share the same physical printer in a round-robin fashion, since the first four bits (0-3) of word "dop" are the same. They both will use the same read compression buffer, print buffer, and UFT (P01). Since these devices have unique disc files (D04, D02), their print files will be separated on the final physical printer even though they are "printed" on the imaginary devices at overlapping times.

10.6.2 CATALOGING A NON-RESIDENT SPOOLER

The spooler will read, write, and store in its own TCB; therefore, the spooler must be a privileged task. The resources for the spooler are specified like any other task; however, some parameters in the resources of the task must be specially prepared.

After entering TOC (Task/Overlay Cataloger), the statements will usually be:

FILE name

name - is the logical file name where the spooler is to be cataloged. This file is usually "SM".

TASK name,level

name - is the spooler task name. This is usually "S" if only one spooler task.

level - is optional but must be specified here if it was not specified on the "PRE-SCHEDULE" statement. This is the priority level the spooler task will run at.

SPACE 0,#3000

This statement is required and says that the spooler has a map size of 12K words (#3000). The spooler will auto automatically allocate its buffers from this space.

PECULIAR PRIVILEGED,SYMBIONT[,AUTOSTART]

PRIVILEGED - says that the spooler will run in the privileged mode. This parameter is required.

SYMBIONT - says that the spooler is a symbiont task. This parameter is required.

AUTOSTART - specifies that the spooler will automatically start-up when the system starts up. This parameter is optional since the spooler will automatically start-up when the first I/O call to it is made.

VARIABLES

This statement is required and specifies that a variables extension (for operator communication) is to be added.

LOGFILE OUT,dev

OUT - is the logical file name where the spooler writes its operator communications, stall messages, et cetera.

dev - is the operator communications terminal where the messages are to be put. This is usually TYn.

Next follows a "LOGFILE" statement for each Dxx spooler disc file and Pxx spooler print file.

Example:

```
LOGFILE D00,ddev
LOGFILE D01,ddev
  .
  .
  .
LOGFILE D15,ddev
LOGFILE P00,pdev
LOGFILE P01,pdev
  .
  .
  .
LOGFILE P15,pdev
```

Dxx - logical file names derived from Bits 4-7 of the DEVICE option word. Refer to Figure 10-1.

ddev - are the disc device names.

Pxx - logical file names derived from Bits 0-3 of the DEVICE option word. Refer to Figure 10-1.

pdev - are the actual printer device names.

IOPERATIONS n

n - is the number of simultaneous I/O operations to be performed. This is usually the total of all the "LOGFILE" statements.

OPTIONS HOLD,LO,GO,SCRATCH

These are the default options for the spooler. Refer to the section entitled "SPOOLER OPTIONS" before choosing these parameters.

CATALOG

Assuming "BI" is already assigned to a device where the spooler object resides, this statement will catalog the spooler on the disc.

10.6.2.1 Standard S Resources

```
EXE  TOC
FILE SM
SOU  S


S-TASK RESOURCES

DEFAULT
FILE  SM
TASK  S      #03 #00
PECULIAR AUTOSTART
IOPERATIONS  #03 #00
STACKS #0064 #000C
SYSPAGES #02
OPTION  NO7
OPTION  NO8
OPTION  NO9
OPTION  NOA
OPTION  NOC
LOGFILE OUT  TY, TY,
LOGFILE D00 SP0,SP0,
LOGFILE P00 LPP,LPP,
ID   1/27/83
PECULIAR PRIVILEGED
PECULIAR SYMBIONT
VARIABLES
ISPACE #0000 #3000
OSPACE #0000 #3000
ASS LO LO
```

10.6.2.2 Change S Resources

```
EXE TOC
FILE SM
ASS BI SCA BO SCA
GET S
SOU S,INIT
LOGFILE P01,LPC,LPC
LOGFILE D01,SP1,SP1
PUT,,NEW
EXI
```

10.6.3 SPECIFYING A RESIDENT SPOOLER

A Resident spooler is not normally used but in some cases memory can be saved by having the spooler resident. A resident spooler is specified similar to a non-resident spooler. However, the user must compute the size of a buffer that is used for compression, reentrant I/O calls, uncompression, et cetera. These statements should read as follows:

TASK S,lvl

- S - is the spooler task name.
- lvl - is the priority level where the spooler will run.

PECULIAR SYMBIONT[,AUTOSTART]

- SYMBIONT - specifies that the spooler is a symbiont task. This parameter is required.
- AUTOSTART - specifies that the spooler will automatically start-up when the system starts up. This parameter is optional since the spooler will start up when the first I/O call to it is made.

VARIABLES

This statement is required and specifies that a variables extension for operator communication is to be added.

LOGFILE OUT,dev

- OUT - is the logical file name where the spooler writes its operator communications, stall messages, et cetera.
- dev - is the operator communications terminal where the messages are able to be put. This is usually TYn.

Next follows a "LOGFILE" statement for each Dxx spooler disc file and Pxx spooler print file.

Example:

```
LOGFILE  D00,ddev
LOGFILE  D01,ddev
  .
  .
  .
LOGFILE  D15,ddev
LOGFILE  P00,pdev
LOGFILE  P01,pdev
  .
  .
  .
LOGFILE  P15,pdev
```

Dxx - logical file names derived from Bits 4-7 of the DEVICE option word. Refer to Figure 10-1.

ddev - are the disc device names.

Pxx - logical file names derived from Bits 0-3 of the DEVICE option word. Refer to Figure 10-1.

pdev - are the actual printer device names.

IOPERATIONS n

n - is the number of simultaneous I/O operations that will be performed. This is usually the total of all the "LOGICAL" statements

OPTIONS HOLD,LO,GO,SCRATCH

These are the default operations for the spooler. Refer to the section entitled "SPOOLER OPTIONS" before choosing these parameters.

FINTASK S

S - is the spooler task name.

The following examples show the statements required in the TASK and I/O of SYSGEN, respectively, to catalog a resident task S:

```
TASK            S,48
PECULIAR        SYMBIONT,AUTOSTART
VARIABLES
LOGFILE         OUT,TY     (operator error and status communication)
LOGFILE         P00,TY     (printer for device (1))*
LOGFILE         P01,LP     (printer for devices (2) and (3))*
LOGFILE         D00,DS1    (disc partition for device (1))*
LOGFILE         D04,DS2    (disc partition for device (2))*
LOGFILE         D02,DS3    (disc partition for device (3))*
IOPERATIONS     6
OPTIONS         LO
FINTASK         S
```

*Refer to examples 1, 2 and 3 in description of DEVICE statement.

```
SYMCONTROLLER   S,100+100+100+100+37+12+100+67+12
```

Assume all partitions are 100 words each, TY is 37 words (74 bytes), and LP is 67 words (134 bytes). The device names and disc partition names assigned above must also be defined with the appropriate system generation statements. Disc partitions should be sized on the basis of expected usage and the size of final printer data to be held.

The TASK statement specifies that S is resident; therefore, a buffer must be specified. The formula for the buffer in the controller table is broken down as follows:

(A) The first group of three 100-word blocks is the set of write compression buffers (the buffers that are needed to take the print information from the caller and compress into, and finally, write to the disc by ths IDS part of the spooler).

(B) The next word block (100+37+12) is needed by the PDS part of the spooler to handle the physical printer assigned to, by P00 (TY). The 100-word portion is for the print compression buffer (the buffer that is needed to read the compressed print information from the disc). The 37-word portion is used to uncompress into and form a printable line and finally write to the printer (TY) from. The 12-word portion is used to build reentrant code (the write to the printer call, the UFT, et cetera). DEVICE (1) above will use this group of buffers.

(C) The last word block (100+67+12) is needed by the PDS part of the spooler to handle the physical printer assigned to, by P01 (LP). The buffers are used for the same purpose as (B) above except that the target printer is now LP instead of TY. DEVICEs (2) and (3) will share this group of buffers.

If this buffer is not properly sized, the spooler will abort at system start-up with the reason code "BUF" (the size needed will be printed on the device assigned to by "OUT").

10.7 SPOOLER OPERATOR DIRECTIVES

The system operator can communicate with the spooler task using the terminal assigned to the OUT file. The operator directives are passed to the spooler from the Operator Communication (OC) task using standard operator directives in any of the following forms:

```
/S/R    xxx,n    \
/S/RES  xxx,n     >  resume spooler and pass the command "xxx,n"
/S/REL  xxx,n    /
/S/LET  xxx,n        pass command to spooler
/S/OPTION O ...O     pass option to the spooler
```

S - is the name of the spooling task.

xxx - is a special command to the spooler.

n - is usually the imaginary-device disc file number specified at system generation time on DEVICE statement parameter 4, ranging from 0 to 15.

The xxx and n parameters may be separated by a comma, a blank, an equal sign, or a slash. The spooler task responds to the following special commands.

10.7.1 HOLD,n

The HOLD directive places the logical device using disc file number n in a hold state. If this file is currently printing, no other file will print unless another file is entered using the PRINT directive. Thus, the physical line printer stops. If this file is not currently printing and a job finishes, this file will not be allocated time to be printed (this file is bypassed by PDS as long as it is held).

10.7.2 RESUME,n

The RESUME directive causes the logical device using file number n to resume. If this file was currently printing when it was held and no other file was told to print, this file starts printing. If this file was not printing when it was held, this file prints when its turn comes in the round-robin file selection scheme. An aiternate form of this command is "R,n".

10.7.3 REPEAT,n

The REPEAT directive causes the logical device using file number n to repeat. If this file is currently printing, all pointers are reset to the beginning of the job and the job restarts. If this file was not printing, all pointers are reset to the beginning of the job and, when this file starts printing, the job starts at the beginning.*

10.7.4 PRINT,n

The PRINT directive causes the logical device using file number n to print. If this file is currently printing or is in the hold state, all pointers are reset to the beginning of the job and the job restarts.* If this file is not printing and is not in the hold state, the file that is printing has its pointer reset to the beginning of its job, and this file (n) starts printing.* The interrupted job (if any) will start printing from the beginning when selected by the "round-robin" selection scheme. An alternate form of this command is "P,n".

10.7.5 KEEP,n

The KEEP directive causes the logical device using disc file number n to keep the entire job. It will not deallocate disc space. If "KEEP" has been specified, all compression buffers associated with this device are always written to the disc without core-to-core bypass and no disc space is deallocated, even if the job has been partially or completely printed.

When a job in the "KEEP" state is completely printed, all pointers are reset to the start of the job. This file is set to the hold state, and the next number in the round-robin chain is selected to print. The operator should "RESUME" or "PRINT" the kept file to obtain multiple copies of the wanted job. When all desired copies have been printed, the operator should "REMOVE" this job. An alternate form of this command is "K,n".

10.7.6 FLUSH,n

The FLUSH directive causes the logical device using disc file number n to deallocate disc space as needed. "FLUSH" removes the "KEEP" status from the specified file. An alternate form of this command is "F,n".

* To avoid lost lines, the scratch option must be on or the file number must have had "KEEP" specified to assure that all the compression buffers are written to the disc.

10.7.7 REMOVE,n

The REMOVE directive causes the logical device using file number n to bypass the physical printing of a job until the end of the job (end-of-file) is detected. This job's disc space is deallocated and the next job is placed in the round-robin queue.

10.7.8 LINES,#nxxx

The LINES directive causes the logical device using file number n to change the number of lines of its imaginary page length. When this number of lines is printed, the EOF (end-of-form) indication is passed to the user in Word 0, Bit 10 of the UFT. Normally, the IDS part of the spooler returns an EOF indication to the user after 46 lines for non-terminal devices or 62 lines for terminal devices. xxx is the number of lines per page wanted. The number must be entered in hexadecimal form.

10,7.9 READ,#nxxx

The READ directive causes the logical device using disc file number n to read from disc sector #xxx. This command should only be used in an emergency situation (like a power failure) to try to recover lost information without having to return or reprint an entire job. #xxx is the relative disc sector of the partition to which the UFT number n is assigned and from which the PDS part of the spooler will read. The number must be entered in hexadecimal form.

10.7.10 WRITE,#nxxx

The WRITE directive causes the logical device using disc file number n to write to disc sector #xxx. This command should be used only in an emergency situation to try to recover lost information. #xxx is the relative disc sector of the partition to which file number n is assigned and to which the IDS part of the spooler will write. The number must be entered in hexadecimal form.

10.7.11 INITIAL,#nxxx

The INITIAL directive causes the logical device using file number n to change the start of the job indication to relative sector #xxx.

10.7.12 abc,n

This directive causes the task device using file n to make an assignment change. If "abc" is a disc partition, the specified disc file UFT (n) is assigned to "abc". If not a disc partition but a legal device, the printer file UFT (n) is assigned to "abc".

10.7.13 STATUS,n and ALL

The STATUS Directive causes the task device using file number n to print on file "OUT" the status of n. The ALL Directive specifies that the status of all spooler devices is wanted.

The format of the status mesage is:

```
                        IU  P  H  K  R
S - D04/DS2/P01/LP/     /  /  /  /  /L 2E/W021E/R015F/I004E/
                        NIU NP NH NK NR
```

S	- is the spooler task name (in case more than one spooler task is running).
D04	- is the disc UFT and 04 is its file number.
DS2	- is the disc partition the spooler spools its listing to for this imaginary device.
P01	- is the printer UFT and 01 is its file number.
LP	- is the physical printer to which the spooler outputs.
IU	- indicates this number (04) is in use (something has been output to this number).
NIU	- indicates this number (04) is not in use (nothing has been output to this number or an end-of-job was found and nothing has been output since).
P	- indicates that this number (04) is printing (this is the number in the round-robin queue that is currently printing).
NP	- indicates that this number (04) is not printing (if IU, this number in the round-robin queue is waiting to be printed or, if NIU, this number has not been output to or an end-of-job was found by PDS and nothing was output to IDS since).
H	- indicates that this number (04) is holding (see "HOLD" command).
NH	- indicates that this number (04) is not holding (see "HOLD" command).
K	- indicates that this number (04) is keeping (see "KEEP" command).

NK - indicates that this number (Ø4) is not keeping (see "KEEP" command).

R - indicates that this number (Ø4) is removing (see "REMOVE" command).

NR - indicates that this number (Ø4) is not removing (see "REMOVE" command).

L 2E - is the number of lines (in hexadecimal) per page (#2E = 46) for this task device number (Ø4) (see "LINES" command).

WØ21E - indicates that relative sector #Ø21E is the next sector to be written on by the IDS part of the spooler for this number (Ø4) (see "WRITE" command).

RØ15F - indicates that relative sector #Ø15F is the next sector to be read by the PDS part of the spooler for this number (Ø4) (see "READ" command).

IØ04E - indicates that relative sector #Ø04E is the start of the job for this number (Ø4) (see "INITIAL" command).

10.8 SPOOLER OPTIONS

The spooler task responds to the setting of the following options:

SC Normally, the spooler tries to take a partially filled compression buffer from the IDS part of the spooler when PDS finds nothing on the disc to process. This means that the IDS compression buffer may not be written to the disc; thus, if a job is to be reprinted, all of that job may not be on the disc. To force the spooler to write all non-terminal task-device compression buffers to the disc, the scratch option of the spooler task should be set. For terminal devices, the disadvantage is that many lines may be processed by the spooler before printing actually occurs. (Non-terminal task devices are specified on the DEVICE statement at system generation without bit 13 of word 4 in the logical device table specified.)

HO Normally, the PDS part of the spooler automatically selects (in a round-robin fashion) the next job to be run when a job finishes (when an end-of-file is detected). If the hold option of the spooler task is set, the PDS part of the spooler suspends the job that has just finished and waits for an operator directive to initiate the next job to be printed.

GO If this option is set and the spooler does not have work to do for about eight seconds, the spooler exits. The spooler is then reloaded by the I/O system when work needs to be done. The spooler must reside on disc and be a prequeued task to use this option. This feature is not recommended in "core-tight" systems where the loading of the spooler might be checkpointed by tasks that use it or might cause checkpointing of tasks that use it.

LO If this option is set, all printer page sizes are initialized to 62 lines. If this option is not set, terminal page sizes are initialized to 62 lines and non-terminal page sizes are initialized to 46 lines. The "LINES" operator directive can override this option.

BO If this option is set the current job being spooled will not be printed until an End-of-File is detected unless the spool file becomes full. This option can be utilized when several tasks are spooling output and only one printer is available, to avoid keeping the printer tied up while a job is in progress.

UO If this option is set, the spooler takes exclusive use of the physical device prior to printing and gives it up at the end of the print file. It is intended for users who have multiple spooler tasks and a limited number of printing devices.

10.9 SPOOLER CARRIAGE CONTROL

The following control characters may be used in byte 0 of a programmer's print data buffer:

?(#3F) This special carriage control byte causes the PDS part of the spooler to print the line on the line printer and on the device to which file "OUT" is assigned, with an "S Dxx" appended to the end of the message line. PDS then holds the file number as specified by xx. The operator should resume this file number to cause printing to resume. This carriage control byte is useful for form change messages, for example. (EOF is returned if the line count expires.)

!(#21) Same as "?" but does not HOLD the file number after the message is output.

blank (#20) The print carriage is upspaced one line before the remaining bytes are printed. (EOF is returned if the line count expires.)

+ (#2B)	The print carriage is not upspaced before printing the remaining bytes. This byte is used to cause overprint. The line count is not incremented; therefore, EOF is not returned.
0 (#30)	The print carriage is upspaced two lines before printing the remaining bytes. (EOF is returned if the line count expires.)
1 (#31)	The print carriage is advanced to top-of-form (or carriage return, four line feeds if a teletype) before printing the remaining bytes. The page count is reset and EOF is not returned.
- (#2D)	This byte is the same as 1 (#31) and is usually used for title lines.
#C0*	Advance to channel 0; same as 1 (#31).
#C8*	Advance to channel 8. The carriage advances to end-of-form before printing of the remaining bytes. (EOF is unconditionally passed to the user.)
#Cx*	Advance to channel x where x is not 0 or 8. The carriage line count is upspaced one line before printing the remaining bytes. (EOF is returned if the line count expires.) The hardware carriage functions that may be specified in channel x are performed.
#8x*	Advance x lines before printing the remainder of the line; x must be in the range of 0-#3F.

* The physical printer must be able to perform these commands with hardware.

CHAPTER 11
MAX IV INPUT SPOOLING SYMBIONT

The input spooling symbiont task (hereafter referred to as RDS) is used to interpose high-speed buffering between a task trying to read from logical device CR, and the actual card reader. Although intended mainly as an input spooler for batch tasks, RDS can also be used to spool input for non-batch tasks under MAX IV. The RDS task can logically be broken down into two basic parts, "physical input" and "imaginary input".

11.1 PHYSICAL INPUT

The physical input section is responsible for reading all cards placed in the card reader(s) and spooling the input on a job-by-job basis to a disc buffer. A job in this context can be defined as input intended for a particular task. This operation is controlled by job cards.

RDS reads all cards placed in the card reader assigned to logical device CRP at system generation. Additional card readers can be added through operator communication directives described below.

11.1.1 JOB CONTROL CARDS

RDS recognizes the following as job control cards:

```
$JOB    (the $ in column 1)
$RDS
$EOJ
```

RDS ignores all cards preceding a $JOB card or following a $EOJ card but preceding the next $JOB. If no $EOJ card is encountered before another $JOB card is read, the $EOJ will be "supplied" by RDS to delineate the input. The $RDS card is not considered part of the input as are $JOB and $EOJ.

11.1.2 $JOB

The job card signals the start of a job to the input spooler. It contains an optional parameter as follows:

```
$JOB [/task/]
```

task - is the 1 to 6 character name of the task for which the input is intended. The forward slashes (/) surrounding the task name are required. Default value: B

Anything else can be put on the $JOB card following the closing forward slash of the task name (or following the $JOB if no task name is given) and the entire card will be passed to a batch task.

11.1.3 $RDS

The $RDS card is used to pass options to RDS. The only option currently defined is alternate card code (Ø26-keypunch) as follows:

```
$RDS U6 or $RDS 6
```

The option is disabled by:

```
$RDS NOU6 or $RDS NO6
```

Standard translation is the default value at the beginning of each job.

11.1.4 $EOJ

The $EOJ card signals the end-of-input for the designated task. It can be left out (it will be "supplied" by RDS) if another $JOB card follows immediately, signaling the start of a new job. The $EOJ card requires no parameters.

11.2 IMAGINARY INPUT

The imaginary input section of the RDS task intercepts all I/O requests directed to logical device CR, fetches the correct information from the disc (if necessary), and directs it to the buffer of the requesting task (in the case of a READ operation). All the operations are transparent to the requesting task, which operates as if reading directly from the card reader.

11.2.1 I/O REQUESTS

All I/O requests directed to logical device CR are "intercepted" by RDS. The symbiont handles the requests in such a manner that to the calling task, it looks just like a card reader. For example, WRITE, BKFILE, BKRECORD, WEOF, and HOME are treated as null operations. READ, AVFILE, and AVRECORD are the only requests that cause the input spooled to the disc to be accessed. A REWIND request zeroes the device position index in the UFT.

11.2.2 SPOOLED INPUT - BATCH TASK

The way the spooled input is passed back to the requesting task depends on whether it is a batch or non-batch task. For a batch task, the input is passed exactly as it was read from the card reader (except for any $RDS cards) including the $JOB and $EOJ cards. The one exception to this rule is if a $EOJ is about to be sent to the batch task, RDS will compare the current overlay name with the task name. If the two names are identical, a $EOJ is sent; otherwise, EOF and EOM are sent as status.

11.2.3 SPOOLED INPUT - NON-BATCH TASK

For a non-batch task, the $JOB and all cards following it with a $ in column 1 will be skipped until the first card without a $ in column 1 is encountered. That card and all cards following it (regardless of what they contain in column 1) are passed intact to the requesting task. The $EOJ card is converted to simulate an empty hopper condition, that is, the EOF and EOM bits will be set in the status word of the UFT.

11.2.4 SPOOLED INPUT EXAMPLES

An example of spooled input to RDS is listed below.

Example:

```
(1)  $JOB /TASKA/
(2)  $ABC
(3)  $123
(4)  ABC
(5)  123
(6)  $EOJ
```

If RDS determines that TASKA is a non-batch task, cards 1-3 will be skipped over when TASKA requests input, cards 4 and 5 will be passed as is and the $EOJ card will be returned as an end-of-file (EOF), end-of-medium (EOM) condition when read. If TASKA is a batch task, cards 1-6 will be passed as is.

Example:

```
(1)  $JOB /TASKB/
(2)  $XYZ
(3)  XYZ
(4)  $XYZ
(5)  DEF
(6)  $EOJ
```

If TASKB is a non-batch task, cards 1 and 2 will be passed over, cards 3-5 will be passed as is, and $EOJ will be passed as an EOM, EOF condition. If TASKB is a batch task, cards 1-6 will be passed to it as is.

This restriction for non-batch tasks is included to allow for future enhancements.

11.3 RDS OPERATOR COMMANDS

The operator can use the "LET" OC directive to command RDS to do the following:

- Display job status
- Remove job from job queue
- Add additional card readers

Each command keyword has a long and short form as described below. Each time "job" is referenced below, it refers to the job identification number assigned by RDS.

11.3.1 DISPLAY JOB STATUS

The STATUS command displays information (on device CO) relating to a job in the symbiont's job queue. If the job identification number is given, the status of only that particular job will be displayed. If the job number is omitted when entering the command, the status of all jobs in the queue is displayed. Job status is automatically displayed when the job is read from the card reader. It should be pointed out that a task can be put in the job queue even if no input was read from the card reader for it. This occurs if the task requests input from device CR. In this special case, the job will be assigned a job number of 0.

```
/RDS/LET STA [job]
/RDS/LET S   [job]
```

The STATUS display consists of five parts that describe the job entry in the RDS job queue as shown below.

```
JOB jjj FLAGS hh ffffffff TASK tttttt TCB aaaa SECTOR ssssss
```

jjj - is the job identification number.
hh - is the hexadecimal display of the job status bits.
ffffffff - is a listing of the job status bits:
- S means stripping (waiting for physical input by stripping I/O requests to the slough queue).
- J means job (input job exists).
- P means physical (now reading cards for this job).
- T means task (the task issued a read before physical input).
- I means imaginary (imaginary device buffer exists).
- U means unstripping (requeuing nodes from slough queue).

tttttt - task name.
aaaa - TCB address of the task (in hexadecimal).
ssssss - job activation data sector number for the job.

11.3.2 REMOVE JOB FROM JOB QUEUE

The REMOVE command causes the specified job to be removed from the job queue. This removal is accomplished by assigning the job's spooled input to RDS, which reads and discards it. Due to this method of removal, the job number remains in the queue for a time.

```
/RDS/LET REM job
/RDS/LET R   job
```

11.3.3 ADD CARD READER(S)

The ALLOCATE command allows the addition of extra card readers for input purposes. To utilize this command, an assignment must be made for the new card reader as file Rnn, where nn ranges from Ø1 to 15. The first digit of the option (r) is the hexadecimal representation of nn. The second digit selects the spooling file to be used. At present, it should always be zero, thus specifying the use of file DOO. The digits i and o specify the number of input and output buffers respectively. Both are usually specified as 1 when using low-speed card readers.

```
/RDS/LET ALL  #rdio
/RDS/LET A    #rdio
```

An example of the ALLOCATE command is shown below for the addition of device CRA as a second spooler input source.

```
/RDS/ASS  RØ1=CRA
/RDS/LET ALL #1Ø11
```

11.4 INSTALLING THE INPUT SPOOLER

The RDS input spooling task operates as a normal MAX IV symbiont. It requires one disc partition for spooling and at least one card reader for input. RDS is designed to be run as a prequeued non-resident task. If the user does not prequeue it in the system generation, the operator will have to activate it using an operator communications directive. The necessary system generation statements and the cataloging procedure necessary to include RDS in a MAX IV system are presented in the following sections.

11.4.1 SYSTEM GENERATION

The assumption is made that the following statement for the card reader controller is already present in the user's SYSGEN:

```
CONTROLLER      CR,5
```

The following statement should be included in the Task Structure Block portion of the SYSGEN to ensure immediate activation of the symbiont when MAX IV is initialized:

```
PRESCHEDULE     RDS,,file
```

file - is the name of the logical file on which RDS is cataloged (usually SM).

To specify the controller for the RDS task, the following statement is required in the logical I/O portion of the SYSGEN:

```
SYMCONTROLLER  RDS
```

To specify that logical device CR be assigned to the symbiont, the following statement is required in the logical I/O portion.

```
DEVICE          CR,RDS,80,#10
```

Since the above statement assigns logical device CR to RDS, it is necessary to include a statement that defines a logical device assigned to the card reader:

```
DEVICE          CRP,CR
```

The user must ensure that a disc partition exists exclusively for spooling input. Assuming the partition is to be called SPC, the following statement suffices:

```
DISCDEVICE      SPC,M0,ntk,,,[stk],RDS
```

ntk - the number of contiguous tracks defining the size of the spooling partition.

stk - when specified, this parameter forces the partition to begin with the specified track, instead of assigned a starting track based on the last previously defined partition.

Examples:

```
DISCDEVICE      SPC,M0,8,,,,RDS
DISCDEVICE      SPC,M0,6,,,130,RDS
```

A summary of the above statements is given below:

```
CONTROLLER      CR,5
SYMCONTROLLER   RDS
DEVICE          CRP,CR
DEVICE          CR,RDS,80,#10
DISCDEVICE      SPC,M0,ntk,,,[stk],RDS
PRESCHEDULE     RDS,,file
```

11.4.2 CATALOGING RDS

After entering TOC (Task-Overlay Cataloger), the statements will usually be:

```
FILE    name
```

name - is the logical file name where the spooler is to be cataloged (usually SM). If prescheduled, must agree with logical file name on PRESCHEDULE SYSGEN statement.

```
TASK   RDS,priority,influence
```

priority - is the priority level of the spooler. RDS should be placed at the lowest priority possible and still be higher priority than the tasks it will spool input for. That is, the spooler's priority level should be equal to or closer to zero than the priority level of the tasks using RDS.

influence - RDS's influence MUST equal or exceed (be closer to zero) the influence of the task it is spooling input to. If the 'influence' parameter is allowed to default it is set equal to the 'priority' parameter.

```
SPACE 0,#1000
```

This statement allocates space for the spooler to use for buffers.

```
PECULIAR PRIVILEGED, SYMBIONT
```

This statement allows RDS to execute in the privileged state and links the symbiont task to its controller table.

```
VARIABLES
```

This statement is required in order to use RDS operator commands.

```
LOGFILE       R00,CRP,CRP
LOGFILE       D00,SPC,SPC
LOGFILE       JC:,CR,CR,
LOGFILE       3
```

R00 is the logical file name used for input from the card reader. D00 is the logical file name used by RDS to spool to the disc and fetch the input back from the disc. The last statement allows the operator to add up to three additional card readers for input.

```
IOPERATIONS 10
```

This statement allows RDS to queue up to 10 I/O operations.

```
CATALOG
```

Assuming BI is already assigned to a device where the RDS object (after link-editing) resides, the CATALOG statement will initiate the cataloging.

The following statements show the method used to catalog the edited RDS task in the standard MAX IV systems:

```
$EXE LIB
ASS BI AL4 BO BSB
GET RDS
WEO BO
EXI
$EXE TOC
ASS BI BSB
FILE SM
TASK RDS #FA #FA
PECULIAR PRIVILEGED, AUTOSTART, SYMBIONT
STACKS #0064 #000C
IOPERATIONS #0A
SYSPAGES #02
BIAS #0000
LOGFILE R00, CRP, CRP
LOGFILE D00, SPC, SPC
LOGFILE JC:, CR, CR,
LOGFILES #03
VARIABLES
ISPACE #0000 #1000
OSPACE #0000 #1000
CAT RDS
```

11.5 EXISTING JOB CONTROL DECKS

Job control decks already prepared for input to batch task B can be used without any changes as long as the last job entered has a $EOJ card at the end. For any other batch tasks, the $JOB cards will have to be updated to include the task name.

APPENDIX A
INTERNAL REPRESENTATIONS OF USASCII CODES

CHAR.	HEX CODE	CHAR.	HEX CODE	CHAR.	HEX CODE	CHAR.	HEX CODE
NUL	00	SPACE	20	@	40	`	60
SOH	01	!	21	A	41	a	61
STX	02	"	22	B	42	b	62
ETX	03	#	23	C	43	c	63
EOT	04	$	24	D	44	d	64
ENQ	05	%	25	E	45	e	65
ACK	06	&	26	F	46	f	66
BEL	07	'	27	G	47	g	67
BS	08	(	28	H	48	h	68
HT	09	)	29	I	49	i	69
LF	0A	*	2A	J	4A	j	6A
VT	0B	+	2B	K	4B	k	6B
FF	0C	,	2C	L	4C	l	6C
CR	0D	-	2D	M	4D	m	6D
SO	0E	.	2E	N	4E	n	6E
SI	0F	/	2F	O	4F	o	6F
DLE	10	0	30	P	50	p	70
DC1	11	1	31	Q	51	q	71
DC2	12	2	32	R	52	r	72
DC3	13	3	33	S	53	s	73
DC4	14	4	34	T	54	t	74
NAK	15	5	35	U	55	u	75
SYN	16	6	36	V	56	v	76
ETB	17	7	37	W	57	w	77
CAN	18	8	38	X	58	x	78
EM	19	9	39	Y	59	y	79
SUB	1A	:	3A	Z	5A	z	7A
ESC	1B	;	3B	[	5B	{	7B
FS	1C	<	3C	\	5C	|	7C
GS	1D	=	3D	]	5D	}	7D
RS	1E	>	3E	^	5E	~	7E
US	1F	?	3F	_	5F	DEL	7F

APPENDIX B
CHARACTER CODES

ASCII		CARD CODE			EBCDIC		COMPRESSED ALPHA NUMERIC		TTY		LINE PRINTER	
ASCII CHAR.	HEX. CODE	029 KEYPUNCH SYMBOL	ANSI CODE		CHAR.	HEX. CODE	COMP. CHAR.	CAN CODE	TTY	HEX. CODE	L.P. SYMBOL	HEX. CODE
NUL	00	M.P.	12-0-9-8-1		NUL	00			NUL	80		
SOH	01	M.P.	12-9-1		SOH	01			SOM	81		
STX	02	M.P.	12-9-2		STX	02			EOA	82		
ETX	03	M.P.	12-9-3		ETX	03			EOM	83		
EOT	04	M.P.	9-7		EOT	37			EDT	84		
ENQ	05	M.P.	0-9-8-5		ENQ	2D			WRU	85		
ACK	06	M.P.	0-9-8-6		ACK	2E			RU	86		
BEL	07	M.P.	0-9-8-7		BEL	2F			BEL	87		
BS	08	M.P.	11-9-6		BS	16			FE	88		
HT	09	M.P.	12-9-5		HT	05			HT	89		
LF	0A	M.P.	0-9-5		LF	25			LF	8A		
VT	0B	M.P.	12-9-8-3		VT	0B			VT	8B		
FF	0C	M.P.	12-9-8-4		FF	0C			FORM	8C		
CR	0D	M.P.	12-9-8-5		CR	0D			RETURN	8D		
SO	0E	M.P.	12-9-8-6		SO	0E			SO	8E		
SI	0F	M.P.	12-9-8-7		SI	0F			SI	8F		
DLE	10	M.P.	12-11-9-8-1		DLE	10			DC0	90		
DC1	11	M.P.	11-9-1		DC1	11			X-ON	91		
DC2	12	M.P.	11-9-2		DC2	12			TAPE	92		
DC3	13	M.P.	11-9-3		DC3	13			X-OFF	93		
DC4	14	M.P.	9-8-4		DC4	3C			TAPE	94		
NAK	15	M.P.	9-8-5		NAK	3D			ERROR	95		
SYN	16	M.P.	9-2		SYN	32			SYC	96		
ETB	17	M.P.	0-9-6		ETB	26			LEM	97		
CAN	18	M.P.	11-9-8		CAN	18			S0	98		
EM	19	M.P.	11-9-8-1		EM	19			S1	99		
SUB	1A	M.P.	9-8-7		SUB	3F			S2	9A		
ESC	1B	M.P.	0-9-7		ESC	27			S3	9B		
FS	1C	M.P.	11-9-8-4		S4	9C			S4	9C		
GS	1D	M.P.	11-9-8-5		GS	1D			S5	9D		
RS	1E	M.P.	11-9-8-6		RS	1E			S6	9E		
US	1F	M.P.	11-9-8-7		US	1F			S7	9F		
SPACE	20	SPACE BAR	(NO PUNCH)		SPACE	40	SPACE	0	SPACE	A0	SPACE	20
!	21	|	12-8-7		!	4F			!	A1	!	21
"	22	"	8-7		"	7F			"	A2	"	22
#	23	#	8-3		#	7B			#	A3	#	23
$	24	$	11-8-3		$	5B	$	39	$	A4	$	24
%	25	%	0-8-4		%	6C			%	A5	%	25
&	26	&	12		&	50			&	A6	&	26
'	27	'	8-5		'	7D			'	A7	'	27
(	28	(	12-8-5		(	4D			(	A8	(	28
)	29	)	11-8-5		)	5D			)	A9	)	29
*	2A	*	11-8-4		*	5C			*	AA	*	2A
+	2B	+	12-8-6		+	4E			+	AB	+	2B
,	2C	,	0-8-3		,	6B			,	AC	,	2C
-	2D	-	11		-	60			-	AD	-	2D
.	2E	.	12-8-3		.	4B	.	3B	.	AE	.	2E
/	2F	/	0-1		/	61			/	AF	/	2F

M.P. = Multi-punch (no symbol).

ASCII		CARD CODE		EBCDIC		COMPRESSED ALPHA NUMERIC		TTY		LINE PRINTER	
ASCII CHAR.	HEX. CODE	029 KEYPUNCH SYMBOL	ANSI CODE	CHAR.	HEX. CODE	COMP. CHAR.	CAN CODE	TTY	HEX. CODE	L.P. SYMBOL	HEX. CODE
0	30	0	0	0	F0	0	27	0	B0	0	30
1	31	1	1	1	F1	1	28	1	B1	1	31
2	32	2	2	2	F2	2	29	2	B2	2	32
3	33	3	3	3	F3	3	30	3	B3	3	33
4	34	4	4	4	F4	4	31	4	B4	4	34
5	35	5	5	5	F5	5	32	5	B5	5	35
6	36	6	6	6	F6	6	33	6	B6	6	36
7	37	7	7	7	F7	7	34	7	B7	7	37
8	38	8	8	8	F8	8	35	8	B8	8	38
9	39	9	9	9	F9	9	36	9	B9	9	39
:	3A	:	8-2	:	7A	:	37	:	BA	:	3A
;	3B	;	11-8-6	;	5E			;	BB	;	3B
<	3C	<	12-8-4	<	4C			<	BC	<	3C
=	3D	=	8-6	=	7E			=	BD	=	3D
>	3E	>	0-8-6	>	6E			>	BE	>	3E
?	3F	?	0-8-7	?	6F			?	BF	?	3F
@	40	@	8-4	@	7C			@	C0	@	40
A	41	A	12-1	A	C1	A	1	A	C1	A	41
B	42	B	12-2	B	C2	B	2	B	C2	B	42
C	43	C	12-3	C	C3	C	3	C	C3	C	43
D	44	D	12-4	D	C4	D	4	D	C4	D	44
E	45	E	12-5	E	C5	E	5	E	C5	E	45
F	46	F	12-6	F	C6	F	6	F	C6	F	46
G	47	G	12-7	G	C7	G	7	G	C7	G	47
H	48	H	12-8	H	C8	H	8	H	C8	H	48
I	49	I	12-9	I	C9	I	9	I	C9	I	49
J	4A	J	11-1	J	D1	J	10	J	CA	J	4A
K	4B	K	11-2	K	D2	K	11	K	CB	K	4B
L	4C	L	11-3	L	D3	L	12	L	CC	L	4C
M	4D	M	11-4	M	D4	M	13	M	CD	M	4D
N	4E	N	11-5	N	D5	N	14	N	CE	N	4E
O	4F	O	11-6	O	D6	O	15	O	CF	O	4F
P	50	P	11-7	P	D7	P	16	P	D0	P	50
Q	51	Q	11-8	Q	D8	Q	17	Q	D1	Q	51
R	52	R	11-9	R	D9	R	18	R	D2	R	52
S	53	S	0-2	S	E2	S	19	S	D3	S	53
T	54	T	0-3	T	E3	T	20	T	D4	T	54
U	55	U	0-4	U	E4	U	21	U	D5	U	55
V	56	V	0-5	V	E5	V	22	V	D6	V	56
W	57	W	0-6	W	E6	W	23	W	D7	W	57
X	58	X	0-7	X	E7	X	24	X	D8	X	58
Y	59	Y	0-8	Y	E8	Y	25	Y	D9	Y	59
Z	5A	Z	0-9	Z	E9	Z	26	Z	DA	Z	5A
[	5B	¢	12-8-2	¢	4A			[	DB	[	5B
\	5C	0-8-2	0-8-2	\	E0			\	DC	\	5C
]	5D	!	11-8-2	!	5A			]	DD	]	5D
^	5E	¬	11-8-7	^	5F			↑	DE	^	5E
_	5F	_	0-8-5	_	6D			←	DF	_	5F

ASCII		CARD CODE		EBCDIC		COMPRESSED ALPHA NUMERIC		TTY		LINE PRINTER	
ASCII CHAR.	HEX. CODE	029 KEYPUNCH SYMBOL	ANSI CODE	CHAR.	HEX. CODE	COMP. CHAR.	CAN CODE	TTY	HEX. CODE	L.P. SYMBOL	HEX. CODE
\	6Ø	M.P.	8-1		79						
a	61	M.P.	12-Ø-1	a	81						
b	62	M.P.	12-Ø-2	b	82						
c	63	M.P.	12-Ø-3	c	83						
d	64	M.P.	12-Ø-4	d	84						
e	65	M.P.	12-Ø-5	e	85						
f	66	M.P.	12-Ø-6	f	86						
g	67	M.P.	12-Ø-7	g	87						
h	68	M.P.	12-Ø-8	h	88						
i	69	M.P.	12-Ø-9	i	89						
j	6A	M.P.	12-11-1	j	91						
k	6B	M.P.	12-11-2	k	92						
l	6C	M.P.	12-11-3	l	93						
m	6D	M.P.	12-11-4	m	94						
n	6E	M.P.	12-11-5	n	95						
o	6F	M.P.	12-11-6	o	96						
p	7Ø	M.P.	12-11-7	p	97						
q	71	M.P.	12-11-8	q	98						
r	72	M.P.	12-11-9	r	99						
s	73	M.P.	11-Ø-2	s	A2						
t	74	M.P.	11-Ø-3	t	A3						
u	75	M.P.	11-Ø-4	u	A4						
v	76	M.P.	11-Ø-5	v	A5						
w	77	M.P.	11-Ø-6	w	A6						
x	78	M.P.	11-Ø-7	x	A7						
y	79	M.P.	11-Ø-8	y	A8						
z	7A	M.P.	11-Ø-9	z	A9						
{	7B	M.P.	12-Ø	{	CØ						
|	7C	M.P.	12-11	|	6A						
}	7D	M.P.	11-Ø	}	DØ						
~	7E	M.P.	11-Ø-1	~	A1						
DEL	7F	M.P.	12-9-7	DEL	Ø7			RUB OUT	FF		

M.P. = Multi-punch (No symbol).

APPENDIX C
CARRIAGE CONTROL FUNCTIONS FOR PRINTING DEVICES

SOFTWARE CARRIAGE CONTROL

SYMBOL	INTERNAL CODE	FUNCTION
SPACE	#20	PRINT AFTER SINGLE UPSPACE
0	#30	PRINT AFTER DOUBLE UPSPACE
1	#31	SKIP TO TOP-OF-FORM (FORM FEED)
+	#2B	SUPPRESS SPACING (ALLOW OVERPRINTING)*
-	#2D	SKIP TO TOP-OF-FORM (INDICATES HEADING THAT SHOULD BE RETAINED)
?	#3F	PRINT AFTER LINE FEED IF DIRECT PRINTER; SUSPEND PRINTING OF FILE FOR FORMS CHANGE IF SPOOLED PRINTER AND OUTPUT MESSAGE TO CONSOLE.
!	#21	PRINT AFTER LINE FEED IF DIRECT PRINTER; DUPLICATE PRINT LINE ON OPERATOR'S CONSOLE OUTPUT DEVICE IF SPOOLED PRINTER

* SOME DEVICES ARE INCAPABLE OF OVERPRINT AND WILL UPSPACE FOR THIS FUNCTION.

HARDWARE CARRIAGE CONTROL

BINARY INTERNAL CODE	FUNCTION
1xxxxxxx	SKIP xxxxxxx LINE FEEDS IF DEVICE NOT CAPABLE OF PERFORMING A HARDWARE FUNCTION.
1100nnnn	SKIP TO CHANNEL nnnn OF CARRIAGE CONTROL TAPE FOR HIGH SPEED PRINTER.
10xxxxxx	SKIP xxxxxx LINE FEEDS FOR HIGH SPEED PRINTER.

APPENDIX D
MODCOMP BINARY CARD FORMAT

TRANSITIONAL CODE	CARD ROWS	INTERNAL ASCII CODE	SYMBOL*	TRANSITIONAL CODE	CARD ROWS	INTERNAL ASCII CODE	SYMBOL*
00	no punch	20	space	40	11	2D	-
01	9	39	9	41	11-9	52	R
02	8	38	8	42	11-8	51	Q
03	8-9	98		43	11-8-9	18	CAN
04	1	31	1	44	11-1	4A	J
05	1-9	91		45	11-1-9	11	DC1
06	1-8	60	\	46	11-1-8	B1	
07	1-8-9	99		47	11-1-8-9	19	EM
08	2	32	2	48	11-2	4B	K
09	2-9	16	SYN	49	11-2-9	12	DC2
0A	2-8	3A	:	4A	11-2-8	5D	]
0B	2-8-9	9A		4B	11-2-8-9	92	
0C	3	33	3	4C	11-3	4C	L
0D	3-9	93		4D	11-3-9	13	DC3
0E	3-8	23	#	4E	11-3-8	24	$
0F	3-8-9	9B		4F	11-3-8-9	8F	
10	4	34	4	50	11-4	4D	M
11	4-9	94		51	11-4-9	9D	
12	4-8	40	@	52	11-4-8	2A	*
13	4-8-9	14	DC4	53	11-4-8-9	1C	FS
14	5	35		54	11-5	4E	N
15	5-9	95		55	11-5-9	85	
16	5-8	27	'	56	11-5-8	29	)
17	5-8-9	15	NAK	57	11-5-8-9	1D	GS
18	6	36	6	58	11-6	4F	O
19	6-9	96		59	11-6-9	08	BS
1A	6-8	3D	=	5A	11-6-8	3B	;
1B	6-8-9	9E		5B	11-6-8-9	1E	RS
1C	7	37	7	5C	11-7	50	P
1D	7-9	04	EOT	5D	11-7-9	87	
1E	7-8	22	"	5E	11-7-8	5E	^
1F	7-8-9	1A	SUB	5F	11-7-8-9	1F	US
20	0	30	0	60	11-0	7D	}
21	0-9	5A	Z	61	11-0-9	7A	z
22	0-8	59	Y	62	11-0-8	79	y
23	0-8-9	88		63	11-0-8-9	B8	
24	0-1	2F	/	64	11-0-1	7E	~
25	0-1-9	81		65	11-0-1-9	9F	
26	0-1-8	B9		66	11-0-1-8	D1	
27	0-1-8-9	89		67	11-0-1-8-9	80	
28	0-2	53	S	68	11-0-2	73	s
29	0-2-9	82		69	11-0-2-9	B2	
2A	0-2-8	5C	\	6A	11-0-2-8	D2	
2B	0-2-8-9	8A		6B	11-0-2-8-9	F4	
2C	0-3	54	T	6C	11-0-3	74	t
2D	0-3-9	83		6D	11-0-3-9	B3	
2E	0-3-8	2C	,	6E	11-0-3-8	D3	
2F	0-3-8-9	8B		6F	11-0-3-8-9	F5	
30	0-4	55	U	70	11-0-4	75	u
31	0-4-9	84		71	11-0-4-9	B4	
32	0-4-8	25	%	72	11-0-4-8	D4	
33	0-4-8-9	8C		73	11-0-4-8-9	F6	
34	0-5	56	V	74	11-0-5	76	v
35	0-5-9	0A	LF	75	11-0-5-9	B5	
36	0-5-8	5F	_	76	11-0-5-8	D5	
37	0-5-8-9	05	ENQ	77	11-0-5-8-9	F7	
38	0-6	57	W	78	11-0-6	77	w
39	0-6-9	17	ETB	79	11-0-6-9	B6	
3A	0-6-8	3E	>	7A	11-0-6-8	D6	
3B	0-6-8-9	06	ACK	7B	11-0-6-8-9	F8	
3C	0-7	58	X	7C	11-0-7	78	x
3D	0-7-9	1B	ESC	7D	11-0-7-9	B7	
3E	0-7-8	3F	?	7E	11-0-7-8	D7	
3F	0-7-8-9	07	BEL	7F	11-0-7-8-9	F9	

TRANSITIONAL CODE	CARD ROWS	INTERNAL ASCII CODE	SYMBOL*	TRANSITIONAL CODE	CARD ROWS	INTERNAL ASCII CODE	SYMBOL*
80	12	26	&	C0	12-11	7C	: or |
81	12-9	49	I	C1	12-11-9	72	r
82	12-8	48	H	C2	12-11-8	71	q
83	12-8-9	97		C3	12-11-8-9	B0	
84	12-1	41	A	C4	12-11-1	6A	j
85	12-1-9	01	SOH	C5	12-11-1-9	A9	
86	12-1-8	A8		C6	12-11-1-8	CA	
87	12-1-8-9	8D		C7	12-11-1-8-9	10	DLE
88	12-2	42	B	C8	12-11-2	6B	k
89	12-2-9	02	STX	C9	12-11-2-9	AA	
8A	12-2-8	5B	[	CA	12-11-2-8	CB	
8B	12-2-8-9	8E		CB	12-11-2-8-9	EE	
8C	12-3	43	C	CC	12-11-3	6C	l
8D	12-3-9	03	ETX	CD	12-11-3-9	AB	
8E	12-3-8	2E	.	CE	12-11-3-8	CC	
8F	12-3-8-9	0B	VT	CF	12-11-3-8-9	EF	
90	12-4	44	D	D0	12-11-4	6D	m
91	12-4-9	9C		D1	12-11-4-9	AC	
92	12-4-8	3C	<	D2	12-11-4-8	CD	
93	12-4-8-9	0C	FF	D3	12-11-4-8-9	F0	
94	12-5	45	E	D4	12-11-5	6E	n
95	12-5-9	09	HT	D5	12-11-5-9	AD	
96	12-5-8	28	(	D6	12-11-5-8	CE	
97	12-5-8-9	0D	CR	D7	12-11-5-8-9	F1	
98	12-6	46	F	D8	12-11-6	6F	o
99	12-6-9	86		D9	12-11-6-9	AE	
9A	12-6-8	2B	+	DA	12-11-6-8	CF	
9B	12-6-8-9	0E	SO	DB	12-11-6-8-9	F2	
9C	12-7	47	G	DC	12-11-7	70	p
9D	12-7-9	7F	DEL	DD	12-11-7-9	AF	
9E	12-7-8	21	!	DE	12-11-7-8	DO	
9F	12-7-8-9	0F	SI	DF	12-11-7-8-9	F3	
A0	12-0	7B	{	E0	12-11-0	BA	
A1	12-0-9	69	i	E1	12-11-0-9	E1	
A2	12-0-8	68	h	E2	12-11-0-8	E0	
A3	12-0-8-9	A7		E3	12-11-0-8-9	C2	
A4	12-0-1	61	a	E4	12-11-0-1	D9	
A5	12-0-1-9	A0		E5	12-11-0-1-9	BB	
A6	12-0-1-8	C3		E6	12-11-0-1-8	D8	
A7	12-0-1-8-9	00	NUL	E7	12-11-0-1-8-9	90	
A8	12-0-2	62	b	E8	12-11-0-2	DA	
A9	12-0-2-9	A1		E9	12-11-0-2-9	BC	
AA	12-0-2-8	C4		EA	12-11-0-2-8	E2	
AB	12-0-2-8-9	E8		EB	12-11-0-2-8-9	FA	
AC	12-0-3	63	c	EC	12-11-0-3	DB	
AD	12-0-3-9	A2		ED	12-11-0-3-9	BD	
AE	12-0-3-8	C5		EE	12-11-0-3-8	E3	
AF	12-0-3-8-9	E9		EF	12-11-0-3-8-9	FB	
B0	12-0-4	64	d	F0	12-11-0-4	DC	
B1	12-0-4-9	A3		F1	12-11-0-4-9	BE	
B2	12-0-4-8	C6		F2	12-11-0-4-8	E4	
B3	12-0-4-8-9	EA		F3	12-11-0-4-8-9	FC	
B4	12-0-5	65	e	F4	12-11-0-5	DD	
B5	12-0-5-9	A4		F5	12-11-0-5-9	BF	
B6	12-0-5-8	C7		F6	12-11-0-5-8	E5	
B7	12-0-5-8-9	EB		F7	12-11-0-5-8-9	FD	
B8	12-0-6	66	f	F8	12-11-0-6	DE	
B9	12-0-6-9	A5		F9	12-11-0-6-9	C0	
BA	12-0-6-8	C8		FA	12-11-0-6-8	E6	
BB	12-0-6-8-9	EC		FB	12-11-0-6-8-9	FE	
BC	12-0-7	67	g	FC	12-11-0-7	DF	
BD	12-0-7-9	A6		FD	12-11-0-7-9	C1	
BE	12-0-7-8	C9		FE	12-11-0-7-8	E7	
BF	12-0-7-8-9	ED		FF	12-11-0-7-8-9	FF	

*NOTE: The codes that do not have a symbol assigned are "non-symbolic" ANSI card codes.

alternate card code, 38, 43, 102
available-sector-list, 79

BACKSPACE, 5
BOF, 59

CARDCODE, 43
carriage control byte, 63, 99
carriage control bytes, 7, 64
carriage control tape, 56
carriage width, 8
carriage width, LPn,LCn,DPn, 57
carriage width, VPn, 64
carriage widths TYn, 4
Centronics printer, 57, 60
communications extensions, 4
compressed code, 47
compression buffer, 79, 98
console teletypewriter, 10
console terminal, 10
CONTROL "A", 6
CONTROL "H", 5
CPn, 47, 51
CRn, 35, 43
CRP, 101

DELETE (DEL), 5
delete character, 5
delete record, 5
Device Dependent Option, PRn, 22
Device Dependent Option, TYn, 12
Disc/Console controllers, 3
DPn, 55, 59
DRn, 35, 43

EMERGENCY QUEUING bit, 10

GE TERMINET, 3, 14

hardware form-feed, 57, 60, 65
Hollerith card code, 38, 47

Imaginary Device Section (IDS), 77
imaginary input section, 102
imaginary spooling device, 77
input spooling symbiont task, 101
intermixed Binary and ASCII, 5, 21, 40
IOIS, 3

job identification number, 104

LCn, 55, 59
LEN, 59

LET, 103
LINEFEED, 4
LINES, 99
lines per page, 96
LPn, 55, 59

Multifunction controller, 3, 60

NUL byte, 47, 48
NULL word, 57, 64

offset function, 48

page length, 59, 96
Physical Device Queue (PDQ), 78
Physical Device Section (PDS), 77
physical input section, 101
PPn, 31
PPx, 31
prefix message, 27
printer page sizes, 99
printing device unbuffered, 60
PRn, 3, 17, 23
PRx, 23

RDS, 101, 102
reason code HOL, 72
reason code SBV, 29, 48
RUBOUT, 5

S, 77
SACI board, 14
SHUTDOWN bit, 78
simultaneous print/plot feature, 67
skip leader, 20
STX, 7
SYMBIONT, 88, 91, 107
SYMCONTROLLER, 85, 93, 105

terminal, 3
terminating byte, 27, 40, 48
TERMINATOR BYTE, 6
TI Silent 700, 3, 14
TOF command, 59
TPn, 3, 31
TPn,PPn, 27
trailing blanks, 5
transition code, 40
transitional code, 47
truncated ASCII, 56
TTx, 31
TYn, 3, 13

unbuffered printing device, 60
uncompress buffer, 81

VPn, 63, 68

WID, 59

XYn, 71, 74

Fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 3624 FT. LAUDERDALE, FL 33309

POSTAGE WILL BE PAID BY ADDRESSEE

MODULAR COMPUTER SYSTEMS
1650 W. McNAB ROAD
P.O. BOX 6099
FT. LAUDERDALE, FLORIDA 33310

Attention: TECHNICAL PUBLICATIONS, M.S. #85

Fold

MODCOMP

Please comment on the publication's completeness, accuracy, and readability. We also appreciate any general suggestions you may have to improve this publication.

If you found errors in this publication, please specify the page number or include a copy of the page with your remarks.

Your comments will be promptly investigated and appropriate action will be taken. If you require a written answer, please check the box and include your address below.

☐

Comments: _____

Manual Title _____

Manual Order Number _____ Issue Date _____

Name _____ Position _____

Company _____

Address _____

_____ Telephone ( ) _____

MCS 148D (1/83)

MODCOMP

Corporate Headquarters:
MODULAR COMPUTER SYSTEMS, Inc., 1650 West McNab Road, P.O. Box 6099, Ft. Lauderdale, FL 33310, Tel: (305) 974-1380, TWX: 310-372-7637

International Headquarters:
MODULAR COMPUTER SERVICES, Inc., The Business Centre, Molly Millars Lane, Wokingham, Berkshire, RG11 2JQ, UK, Tel: 0734-786808, TLX: 851849149

Latin American Sales Headquarters:
MODULAR COMPUTER SYSTEMS, Inc., 1650 West McNab Road, P.O. Box 6099, Ft. Lauderdale, FL 33310, Tel: (305) 975-6562, TLX: 3727652

Canadian Headquarters:
MODCOMP Canada, Ltd., 400 Matheson Blvd. East, Unit 24, Mississauga, Ontario, Canada L4Z 1N8, Tel: (416) 890-0666, TELEX: 06-961279

SALES & SERVICE LOCATIONS THROUGHOUT THE WORLD

"The technical contents of this document, while accurate as of the date of publication, are subject to change without notice."

Printed in the U.S.A.